



(19)  
Bundesrepublik Deutschland  
Deutsches Patent- und Markenamt

(10) DE 694 32 726 T2 2004.04.15

(12) **Übersetzung der europäischen Patentschrift**

(97) EP 0 631 239 B1

(51) Int Cl.<sup>7</sup>: G06F 13/36

(21) Deutsches Aktenzeichen: 694 32 726.3

(96) Europäisches Aktenzeichen: 94 107 528.5

(96) Europäischer Anmeldetag: 16.05.1994

(97) Erstveröffentlichung durch das EPA: 28.12.1994

(97) Veröffentlichungstag

der Patenterteilung beim EPA: 28.05.2003

(47) Veröffentlichungstag im Patentblatt: 15.04.2004

(30) Unionspriorität:

11286093 14.05.1993 JP

(84) Benannte Vertragsstaaten:

DE, FR, GB

(73) Patentinhaber:

Matsushita Electric Industrial Co., Ltd., Kadoma,  
Osaka, JP

(72) Erfinder:

Iitsuka, Hiroyuki, Katano-shi, Osaka, JP;  
Nishimura, Takuya, Katano-shi, Osaka, JP

(74) Vertreter:

Eisenführ, Speiser & Partner, 28195 Bremen

(54) Bezeichnung: Verfahren und System zur seriellen Datenübertragung

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

## Beschreibung

### Hintergrund der Erfindung

#### 1. Gebiet der Erfindung

[0001] Die vorliegende Erfindung betrifft ein Datenübertragungsverfahren zum Übertragen von Datenframes bzw. Datenblöcken zwischen mehreren Einrichtungen wie Ein-Chip-Mikrocomputern (im folgenden „Mikrocomputer“ genannt) unter Verwendung eines seriellen Datenbus mit drei Signalleitungen, nämlich einer Datenleitung, einer Taktleitung und einer Steuerleitung.

#### 2. Stand der Technik

[0002] Die Datenübertragungsverfahren lassen sich grob in parallele Übertragungsverfahren unter Verwendung eines parallelen Datenbus und serielle Übertragungsverfahren unter Verwendung eines seriellen Datenbus (im folgenden „serieller Bus“ genannt) einteilen. Während sich die parallele Übertragung gut für den Datentransfer mit hoher Geschwindigkeit eignet, ist bei diesem Verfahren auch eine größere Anzahl von Anschlussstiften in den Übertragungseinrichtungen, d. h. Mikrocomputern und ICs, erforderlich. Während die seriellen Übertragungsverfahren langsamer sind als die parallelen Verfahren, sind dafür weniger Anschlussstifte erforderlich, und die Übertragungsgeschwindigkeit ist für viele Mikrocomputeranwendungen, einschließlich der Mikrocomputer, die in Audio- und Bildgeräten sowie in Haushaltsgeräten verwendet werden, für die typischerweise serielle Übertragungsverfahren eingesetzt werden, ausreichend. Durch die Zunahme der Funktionen von Verbraucherelektronik wie Videokassettenrecordern und Camcordern sind in einem Gerät oft mehrere Mikrocomputer eingebaut.

[0003] In derartigen Produkten arbeiten die Mikrocomputer nicht unabhängig voneinander, sondern durch Datenaustausch über serielle Busse. Zusätzlich zu den Mikrocomputern können an diesen seriellen Bus noch verschiedene andere IC- und LSI-Elemente angeschlossen sein.

[0004] Ein Datenübertragungsverfahren, bei dem ein serieller Bus mit drei Leitungen verwendet wird, ist im JP 4-332065 beschrieben. Das dort beschriebene Datenübertragungsverfahren verwendet zusätzlich zu der Datenleitung und der Taktleitung eine Steuerleitung zur Identifikation der übertragenen Daten als Sendeberechtigung oder Datenübertragungsblock. Der als Steuereinrichtung fungierende Mikrocomputer gibt nacheinander eine Sendeberechtigung an jeden mit dem seriellen Bus verbundenen Mikrocomputer aus, wenn die Steuerleitung den Pegel High hat. Wenn ein Mikrocomputer die Sendeberechtigung empfängt, die Zugangsrechte zum seriellen Bus erhält und zum Übertragen eines Datenblocks über den Bus ansetzt, schaltet der Mikrocomputer die Steuerleitung auf LOW, um zum Hauptrechner zu werden und den anderen Mikrocomputern seine Nutzung des seriellen Busses anzuzeigen. Wenn die Steuerleitung in den L-Pegel wechselt, beendet die Steuerung die Ausgabe der Sendeberechtigung und gibt den für die Datenblockübertragung erforderlichen Takt aus. Nach diesem Takt beginnt der Hauptrechner mit der Datenblockübertragung. Nach Beendigung der Datenblockübertragung setzt die Steuerung die Steuerleitung wieder auf H-Pegel. Die Steuerung erkennt somit, wann die Steuerleitung wieder den H-Pegel hat, um mit der Ausgabe der Sendeberechtigung zu beginnen.

[0005] Dieses Datenübertragungsverfahren ermöglicht es, Datenkollisionen am Bus zu verhindern, erfordert keine Modulation für die Übertragung und ermöglicht es, Datenblöcke von beliebigem Muster zu übertragen. Die Nachteile dieses Datenübertragungsverfahrens werden nachstehend beschrieben.

[0006] Bei diesem bekannten Datenübertragungsverfahren gibt die Steuerung eine Sendeberechtigung aus, und nur der Mikrocomputer, der die Sendeberechtigung empfangen hat, hat das Recht, einen Datenblock zu senden. Wenn ein weiterer Mikrocomputer hinzukommt, der Datenblöcke über den seriellen Bus überträgt, muss das Sendeberechtigungserteilungsprogramm der Steuerung folglich so geschrieben sein, dass die Möglichkeit einer Vergrößerung der Anzahl der Mikrocomputer, die zum Hauptrechner werden können, berücksichtigt wird, und das Sendeberechtigungserteilungsprogramm der Steuerung muss geändert werden, wenn Hauptrechner, die nicht berücksichtigt wurden, hinzukommen.

[0007] Bei dem bekannten Datenübertragungsverfahren sind der oder die Nebenrechner, die die übertragenen Daten empfangen sollen, ebenfalls durch einen Adresskopf am Anfang des Datenblocks gekennzeichnet, und die Blockübertragung ist beendet, wenn der oder die Nebenrechner eine Bestätigung ausgegeben haben. Folglich müssen auch die Adressdaten geändert werden, wenn ein weiterer zum Nebenrechner bestimmter Mikrocomputer an den seriellen Bus angeschlossen wird, und eine Datenübertragung ist nicht möglich, wenn der Nebenmikrocomputer ohne Adressänderung entfernt wird. Kurz gesagt, ein Problem des bekannten Datenübertragungsverfahrens liegt darin, dass Mikrocomputer nicht ohne weiteres an den seriellen Bus angeschlossen oder von ihm getrennt werden können.

[0008] Mit dem Problem des einfachen Anschließens von Mikrocomputern an einen seriellen Dreileitungsbus und ihrer einfachen Entfernung beschäftigt sich das JP-63-58567 „Serial Interface Serial Bus Method“. Bei die-

sem Verfahren wird der N-Zustand der Steuerleitung bestätigt, die Steuerleitung dann auf LOW geschaltet, eine Kennung gesendet, und die Datenübertragung erfolgt, wenn die empfangene Kennung zur übertragenen Kennung passt. Wenn die Datenübertragung beendet ist, wird die Steuerleitung wieder auf H-Pegel geschaltet. Weil dieses Verfahren keine Entscheidung unter Verwendung von Sendeberechtigungen kennt, können Mikrocomputer ohne weiteres an den seriellen Bus angeschlossen und wieder von ihm getrennt werden. Es ist jedoch auch möglich, dass mehrere Mikrocomputer gleichzeitig eine Kennung senden, was zu Kollisionen führt. Folglich ist es nicht möglich, die Zeit vom Datensendeauftrag bis zur tatsächlichen Beendigung der Datenübermittlung in einem bestimmten Zeitabschnitt zu halten. Mit zunehmender Datenübermittlungsfrequenz steigt auch die Anzahl der Kollisionen sowie die Möglichkeit, dass keine wirksame Datenübermittlung aufrechterhalten werden kann.

[0009] Bei dem ersten herkömmlichen Datenübermittlungsverfahren gemäß dem vorgenannten JP-4-332065 müssen die nicht zur Steuerung gehörenden Mikrocomputer (nachstehend „Targets“ genannt) auch die Steuerleitung mit externen Unterbrechungen überwachen, wenn eine effektive Datenübertragung stattfinden soll. Wenn die Targets den Steuerleitungspegel bei seriellen Empfangsunterbrechungen statt bei externen Unterbrechungen überprüfen sollen, kann es zu Pegelerfassungsfehlern kommen, wenn der Steuerleitungspegel in der Zeit von der Ausgabe der seriellen Empfangsunterbrechung bis zur Pegelüberprüfung vom Hauptrechner geändert wird.

[0010] Ein mögliches Verfahren zur Vermeidung solcher Erfassungsfehler ist es, den Hauptrechner eine bestimmte Zeit nach Beendigung der Übertragung einer Übertragungseinheit warten zu lassen, bevor die Steuerleitung betätigt wird, so dass die Mikrocomputer neben dem Hauptrechner die Möglichkeit haben, die Steuerleitung eine vorgegebene Zeit lang nach Ausgabe der seriellen Empfangsunterbrechung zu prüfen. Kurz gesagt, werden Steuerleitungspegelerfassungsfehler vermieden, indem die zeitliche Abfolge der Steuerleitungsfunktionen gesteuert wird.

[0011] Weil bei diesem Verfahren die Steuerleitung während eines festgelegten Zeitabschnitts kontrolliert werden muss, müssen jedoch serielle Empfangsunterbrechungen auch bei Verarbeitung anderer Unterbrechungen berücksichtigt werden. Zusätzlich ist entweder ein Hardware- oder ein Software-Zeitgeber erforderlich, um die zeitliche Abfolge auf der Steuerleitung zu regeln. Folglich müssen die Targets die Steuerleitung unter Verwendung externer Unterbrechungen überwachen, weil Verfahren zur Verhinderung von Erfassungsfehlern durch Einschränkung der zeitlichen Abfolge von Steuerleitungsabläufen zahlreiche Einschränkungen für die Systemgestaltung bedeuten.

[0012] Das andere Problem bei diesem herkömmlichen Datenübertragungsverfahren (siehe das JP-1992-332065) liegt in der schlechten Datenübertragungseffizienz, weil verschiedene Arten von Steuerdaten (z. B. Sendeberechtigungen, Adressdaten, Fehlererfassungs-codes und Bestätigungsdaten) als Teil des Übermittlungsprotokolls bei jeder Datenblockübermittlung zusätzlich übermittelt werden müssen.

#### Zusammenfassung der Erfindung

[0013] Eine Aufgabe der vorliegenden Erfindung ist es somit, ein Datenübermittlungsverfahren zur Verfügung zu stellen, bei dem a) während der Datenblockübermittlung keine Übermittlungskollisionen vorkommen, b) Mikrocomputer ohne weiteres an den seriellen Bus angeschlossen und wieder von ihm getrennt werden können, c) keine externen Unterbrechungen zur Überwachung der Steuerleitung erforderlich sind und d) die Datenübermittlungseffizienz gesteigert werden kann.

[0014] Die genannten Aufgaben werden gemäß einem ersten Aspekt der Erfindung gelöst durch Bereitstellung eines Datentransferverfahrens zum seriellen Übertragen von Datenframes zwischen einer Steuereinrichtung und einem oder mehreren Targets, welche wechselseitig mit einem Signalbus verbunden sind, der eine bidirektionale Datenleitung zum seriellen Übertragen von zumindest den Daten, die einen Datenframe enthalten, eine Taktleitung zum Zuführen eines Sendetakts für die von der Steuereinrichtung über die Datenleitung zu den Targets übertragenen Daten und eine bidirektionale Wired-or-Steuerleitung aufweist, wobei das Verfahren dadurch gekennzeichnet ist, dass die Steuereinrichtung damit beginnt, einen Datenframe oder einen Nullframe zu der Datenleitung in Abhängigkeit davon zu übertragen, ob eine Datenübertragungsanforderung für die Steuereinrichtung vorliegt oder nicht, nach der Ausgabe eines LOW-Pegels an die Steuerleitung, wenn die Steuerleitung den Pegel HIGH aufweist, und dass die Steuereinrichtung die Ausgabe des Pegels LOW an die Steuerleitung anhält, nachdem das Übertragen des Datenframes oder des Nullframes beendet worden ist, dass die Übertragungsfolge für die Targets vorbestimmt ist, dass die Targets mit einer Übertragungsanforderung für einen Datenframe beginnen, einen LOW-Pegel an die Steuerleitung auszugeben, während die Steuerleitung auf dem Pegel LOW durch die Steuereinrichtung gehalten wird, dass, falls die Steuerleitung auf dem Pegel LOW bleibt, sogar wenn die Steuereinrichtung die Ausgabe des Pegels LOW an die Steuerleitung anhält, nachdem das Übertragen des Datenframes oder des Nullframes geendet hat, die Steuereinrichtung den Sendetakt ausgibt, bis die Steuerleitung den Pegel HIGH annimmt, und das Target, welches einen LOW-Pegel an die Steuerleitung ausgibt, die Übertragung eines Datenframes in Synchronisation mit dem Sendetakt be-

ginnt, nachdem die Steuereinrichtung und alle Targets, die in der Übertragungsfolge früher liegen, das Datenframe- oder Nullframeübertragen beendet haben, und die Ausgabe des LOW-Pegels an die Steuerleitung beendet, nachdem die Datenframeübertragung beendet ist.

[0015] Gemäß einem zweiten Aspekt der Erfindung wird ferner ein Datentransfersystem zur Verfügung gestellt, enthaltend eine Vielzahl an Mikrocomputern, von denen einer als Steuereinrichtung bezeichnet wird und die übrigen als Targets bezeichnet werden, und einen seriellen Datenbus, der eine bidirektionale Datenleitung enthält, über die ein Senden und ein Empfangen von Daten ausgeführt wird, eine Taktleitung, welche Taktsignale, die durch die Steuereinrichtung erzeugt werden, an jedes Target überträgt, und eine bidirektionale Wired-or-Steuerleitung, welche sowohl für die Steuereinrichtung als auch für die Targets zugreifbar und ablehnbar ist, dadurch gekennzeichnet, dass die Steuereinrichtung einen Datenframe oder einen Nullframe an die Datenleitung in Abhängigkeit davon überträgt, ob eine Datenübertragungsanforderung für die Steuereinrichtung besteht oder nicht, nachdem die Steuereinrichtung auf die Steuerleitung zugreift, wenn die Steuerleitung den Pegel HIGH annimmt, und, falls die Steuerleitung auf dem Pegel LOW sogar dann bleibt, wenn die Steuereinrichtung die Ausgabe des Pegels LOW an die Steuerleitung unterbricht, nachdem das Übertragen des Datenframes oder des Nullframes beendet worden ist, die Steuereinrichtung das Taktsignal ausgibt, bis die Steuerleitung den Pegel HIGH annimmt, und dass jedes Target, das eine eigene, ihm zugeordnete Datensendefolge besitzt, eine Ausgabe für das Zugreifen auf die Steuerleitung ausgibt, während die Steuerleitung in einem Zugangszustand durch die Steuereinrichtung gehalten wird, falls eine Sendeanforderung eines Datenframes besteht, und, wenn die Steuereinrichtung und die vorgenannten Targets das Datenframe- oder Nullframeübertragen beendet haben, eine Datenframeübertragung in Synchronisation mit den Taktsignalen von der Steuereinrichtung beginnt und, falls es keine Datenframeübertragungsanforderung besitzt, einen Nullframe an die Datenleitung ausgibt, ohne dass eine Ausgabe zum Zugreifen auf die Steuerleitung ausgegeben wird, und wenn das Frameübertragen beendet worden ist, die Ausgabe für den Zugriff auf die Steuerleitung unterbricht.

[0016] Weitere zweckmäßige Ausführungsbeispiele sind in den abhängigen Ansprüchen definiert.

[0017] Mit dem erfindungsgemäßen Datenübertragungsverfahren werden somit Daten übertragen, wobei an den seriellen Bus angeschlossene Einrichtungen erfindungsgemäß als Steuereinrichtungen oder Targets festgelegt sind. An dem seriellen Bus ist nur eine Steuereinrichtung vorhanden, während mehrere Targets vorhanden sein können. Die Framesendefolge ist festgelegt, und die Steuereinrichtung steht in der Sendefolge an erster Stelle.

[0018] Die Steuereinrichtung und ein oder mehrere Targets sind an einen seriellen Bus angeschlossen, der drei Signalleitungen hat, nämlich eine Datenleitung, eine Taktleitung und eine Steuerleitung. Die Datenleitung ist eine bidirektionale Signalleitung zum seriellen Übertragen der Daten mit dem Datenframe in einzelnen Übertragungseinheiten. Die Taktleitung ist ein Signal zur intermittierenden Bereitstellung des Einheitstakts, der zum Übertragen einer Datenübertragungseinheit erforderlich ist, zu vorgegebenen Zeitpunkten von der Steuereinrichtung an das oder die Targets. Die Steuerleitung ist eine bidirektionale Signalleitung, die zur Meldung des Datenübertragungsbeginns und – endes verwendet wird.

[0019] Nahstehend wird das erfindungsgemäße Datenübertragungsverfahren beschrieben.

[0020] Wenn die Steuerleitung auf H-Pegel ist, beginnt die Steuereinrichtung die Ausgabe des L-Pegels an die Steuerleitung und überträgt dann einen Datenframe oder einen Nullframe. Targets mit Datenblocksendeanforderung beginnen die Ausgabe eines L-Signals an die Steuerleitung, während die Steuerleitung von der Steuereinrichtung auf LOW gehalten wird. Die Steuereinrichtung beendet zeitweise die Ausgabe des L-Pegels an die Steuerleitung zjm Zeitpunkt der Einheitstaktausgabe, nachdem das Senden des Datenframes oder des Nullframes beendet ist.

[0021] Wenn die Steuerleitung auf LOW bleibt, obwohl die L-Pegel-Ausgabe beendet ist, wird der Einheitstakt ausgegeben, nachdem die L-Pegel-Ausgabe an die Steuerleitung wieder eingesetzt hat. Das Target, welches einen L-Pegel an die Steuerleitung ausgibt, beginnt die Datenframeübermittlung synchron zum Einheitstakt, nachdem die Steuereinrichtung und alle in der Sendefolge an früherer Stelle stehenden Targets das Nullframe- oder Datenframesenden beendet haben. Das Target, welches das Datenframesenden beendet hat, beendet die L-Pegel-Ausgabe an die Steuerleitung vor dem Ausgabezeitpunkt des nächsten Einheitstakts.

[0022] Die Steuereinrichtung berechnet für alle während des L-Pegels der Steuerleitung über die Datenleitung übertragenen Daten einen Fehlererfassungscode und übermittelt diesen, nachdem die Steuerleitung den H-Pegel angenommen hat. Alle Targets erfassen Übertragungsfehler unter Verwendung der von ihnen über die Datenleitung gesendeten oder empfangenen Daten und des empfangenen Fehlererfassungscode. Wird ein Fehler festgestellt, wird ein Anforderungscode, bei dem mindestens ein Fehlererfassungsbit auf TRUE gesetzt ist, übermittelt.

[0023] Eine Übertragungseinheit ist definiert als der Zeitabschnitt von der ersten L-Pegel-Einstellung der Steuerleitung bis zu ihrem nächsten L-Pegel. Eine Übertragungseinheit umfasst daher sowohl eine LOW- als auch eine HIGH-Phase der Steuerleitung. Während einer Übertragungseinheit können mehrere Datenblöcke gesendet werden.

[0024] Mit dem oben beschriebenen erfindungsgemäßen Verfahren ist unter Beachtung der vorgegebenen

Sendefolge eine kollisionsfreie Datenblockübermittlung durch alle Mikrocomputer mit Sendeauftrag während der LOW-Phase der Steuerleitung möglich. Dieses Verfahren hat insbesondere den Effekt, dass Datenblockkollisionen verhindert werden, indem jedem an den seriellen Bus angeschlossenen Mikrocomputer eine gesonderte Sendefolgenummer zugeordnet wird.

[0025] Wenn die Systemkonfiguration, d. h. die Anzahl der an den seriellen Bus angeschlossenen Mikrocomputer, geändert wird, indem einer oder mehrere Mikrocomputer entfernt oder hinzugefügt werden, kann die Datenübermittlung in vollem Umfang aufrechterhalten werden, indem sichergestellt wird, dass jedem Mikrocomputer am Bus eine gesonderte Sendefolgenummer zugeordnet wird. Folglich ist es nicht nötig, das Verwaltungsprogramm der Steuereinheit für den seriellen Bus zu ändern, sobald die Mikrocomputerkonfiguration geändert wird.

[0026] An den Beginn eines Datenblocks wird auch ein ihn kennzeichnender Datenkopf gesetzt. Jeder an den Bus angeschlossener Mikrocomputer kann dann feststellen, ob der aktuelle Datenblock zu empfangen ist, indem er den Datenkopf liest. Genauer gesagt wird jedem Datenblock nur ein Datenkopf hinzugefügt, während in dem Datenblock keine Adressdaten enthalten sind, die den Bestimmungsort des Datenblocks kennzeichnen. Daher ist es nicht nötig, den Datenblock zu ändern, wenn die Mikrocomputerkonfiguration geändert wird. Daher ist es mit der Erfindung ganz einfach, zusätzliche Mikrocomputer an den seriellen Bus anzuschließen oder wieder davon zu trennen.

[0027] Außerdem ändert die Steuerung den Steuerleitungspegel unmittelbar vor Ausgabe des Einheitstakts, und die Targets prüfen den Steuerleitungspegel unmittelbar nach Einheitstakteingabe und beginnen und beenden die L-Pegel-Ausgabe an die Steuerleitung. Im einzelnen geben die Targets nur dann einen L-Pegel an die Steuerleitung aus, wenn die Steuerleitung auf LOW ist, so dass Pegeländerungen der Steuerleitung auf den Zeitpunkt unmittelbar vor Ausgabe des Einheitstakts beschränkt sind. Daher ist eine exakte Erfassung des Steuerleitungspegels möglich, wenn die Targets den Steuerleitungspegel unmittelbar nach Softwareeingabe des Einheitstakts kontrollieren. Die Erfindung hat daher den weiteren Effekt, dass sie keine externen Unterbrechungen zur Überwachung der Steuerleitung erforderlich macht.

[0028] Die Steuerung berechnet außerdem den Fehlererfassungscode für alle während des LOW-Zustands der Steuerleitung über die Datenleitung gesendeten Daten unter Verwendung der von der Steuerung über die Datenleitung gesendeten und empfangenen Daten und sendet den Fehlererfassungscode, nachdem die Steuerleitung den H-Pegel angenommen hat. Alle Targets erfassen Übertragungsfehler unter Verwendung der von ihnen über die Datenleitung gesendeten oder empfangenen Daten und des jeweils empfangenen Fehlererfassungscode. Wenn ein Fehler festgestellt wird, wird ein Anforderungscode, bei dem mindestens ein Fehlererfassungsbit auf TRUE gesetzt ist, zu einem vorgegebenen Zeitpunkt gesendet. Daher muss nicht jedem Datenblock ein Fehlererfassungscode angefügt werden, und die Fehlererfassungsfähigkeit dieses Verfahrens ist dennoch ungefähr gleich gut wie bei Datenübertragungsverfahren, bei denen jedem Datenblock ein Fehlererfassungscode angefügt ist. Ferner ist es nicht nötig, jeden empfangenen Datenblock zu bestätigen, und es kann eine effiziente Datenblockübertragung erreicht werden.

[0029] Ein Fehlererfassungscode zum Erfassen von Übertragungsfehlern im Datenkopf ist ebenfalls in den Datenkopf eingefügt. Wenn das Target im Datenkopf einen Übertragungsfehler feststellt, wird die Ausgabe gestoppt, wenn gerade der L-Pegel an die Steuerleitung ausgegeben wird. Damit wird die Übertragung sofort bei Fehlererkennung beendet. Die Erfindung bewirkt also außerdem, dass eine unnütze Datenübertragung unterbleibt.

[0030] Wenn ein von der Steuerung und dem Target zu empfangender Datenblock nicht empfangen wird, wird ein Anforderungscode gesendet, bei dem mindestens ein Bit auf TRUE gesetzt ist. Wenn das Datenblockanforderungsbit des empfangenen Anforderungscodes TRUE ist, fahren die Steuerung und alle Targets mit der Übertragung aller Datenblöcke fort, die von der jeweiligen Einrichtung zu senden sind, bis das Anforderungsbit im Anforderungscode FALSE ist. Die Erfindung bewirkt somit außerdem, dass verlorengegangene Daten empfangen werden können, selbst wenn der bisher empfangene Datenblock durch Auftreten einer Störung, beispielsweise einer Rücksetzung, verlorengegangen ist.

[0031] Wenn das Datenblockanforderungsbit stets TRUE ist, bleibt die Einrichtung ständig im Datensendestatus. Wenngleich das erfindungsgemäße Datenübertragungsverfahren ereignisgesteuert ist, ist es also auch gut anpassbar an Systeme, die zur kontinuierlichen Datenblockübermittlung verwendet werden. Wenn ein serieller Bus für den kontinuierlichen Transfer in einen ereignisgesteuerten Transferbus geändert wird, muss sowohl das Kommunikationsprogramm, welches das Senden und das Empfangen von Datenblöcken steuert, als auch das Anwendungsprogramm, das bestimmt, welche Datenblöcke zu senden sind, geändert werden. Das ist erforderlich, weil das Anwendungsprogramm bestimmt, ob Änderungen am Inhalt der gesendeten Datenblöcke gemacht wurden. Wenn das erfindungsgemäße Datenübertragungsverfahren in Systeme für den kontinuierlichen Datentransfer eingebracht wird, ist jedoch keine Änderung des Anwendungsprogramms erforderlich, wenn das Datenblockanforderungsbit stets TRUE ist.

[0032] In dem vom Target gesendeten Datenkopf ist auch die Datenlänge der Userdaten angegeben. Somit sendet die Steuerung einen Anforderungscode mit einem auf TRUE gesetzten Fehlererfassungsbit, wenn die

Steuerleitung wieder auf HIGH gesetzt wird, bevor die angegebene Userdatenlänge vom Target gesendet worden ist. Die Erfindung ermöglicht es somit, eine Datenblockübertragung unwirksam zu machen, wenn das Target, welches den Datenblock sendet, zurückgesetzt wird.

[0033] Ferner ermöglicht die vorliegende Erfindung eine sichere Datenblockübertragung selbst bei Schwankungen der Anlaufzeit nach der Rücksetzung, indem ein Anforderungscode gesendet wird, der das Fehlererfassungsbit vom Target auf TRUE setzt, wenn die Steuerleitung beim Anlaufen auf L-Pegel ist.

[0034] Wenn die Steuerleitung beim Anlaufen auf LOW ist, wird außerdem der Takt an die Taktleitung ausgegeben, bis die Steuerleitung HIGH wird. Nachdem die Steuerleitung den H-Pegel angenommen hat, sendet die Steuerung einen Anforderungscode mit einem auf TRUE gesetzten Fehlererfassungsbit. Wenn die Steuerleitung beim Anlaufen H-Pegel hat, sendet die Steuerung einen Anforderungscode mit einem auf TRUE gesetzten Fehlererfassungsbit mindestens zweimal, bevor sie den L-Pegel an die Steuerleitung ausgibt. Die Erfindung ermöglicht somit ferner, Datenübermittlungen während der Transfereinheit bedingt zu stornieren, wenn die Steuerung mitten in einer Transfereinheit zurückgesetzt wird.

[0035] Der Effekt der Einheitstaktausgabe nach Bestätigung durch die Steuerung, dass das Target die Datenübertragungsvorbereitungen beendet hat, kann ferner durch Ausführung folgender Prozedur erzielt werden: Das den Datenblock sendende Target ändert die Datenleitung auf den L-Pegel, wenn es die Vorbereitungen zum Senden der einen Übertragungseinheit von Daten, die den Datenblock enthalten abgeschlossen hat. Die Steuerung gibt den Einheitstakt aus, wenn die Datenleitung unmittelbar vor Ausgabe des Einheitstakts für die Datenblockübermittlung durch das Target auf LOW ist. Die Steuerung kann also den Einheitstakt ausgeben, nachdem sie bestätigt hat, dass das Target die Datensendevorbereitungen beendet hat, indem die Datenleitung gemäß dieser Prozedur verwendet wird.

#### Kurzbeschreibung der Zeichnungen

[0036] Die Erfindung wird besser verständlich durch die nachfolgende ausführliche Beschreibung und die beigefügten Diagramme, die folgendes zeigen:

[0037] **Fig. 1** ein Blockschaltbild eines Datenübertragungssystems, welches die bevorzugte Ausführungsform des erfindungsgemäßen Datenübertragungsverfahrens verwendet;

[0038] **Fig. 2** ein Taktfolgediagramm, das die Beziehung zwischen seriellen Daten, Takten und /BSY-Leitung bei der bevorzugten Ausführungsform zeigt;

[0039] **Fig. 3** das Kopfformat bei der bevorzugten Ausführungsform;

[0040] **Fig. 4** das Datenblockformat bei der bevorzugten Ausführungsform;

[0041] **Fig. 5** die Beziehung zwischen den Datenkopfwerten und der Anzahl Bytes N in den Userdaten;

[0042] **Fig. 6** das Format des REQ-Bytes bei der bevorzugten Ausführungsform;

[0043] **Fig. 7** ein Taktfolgediagramm, das ein Beispiel für den Datenblockübertragungstakt bei der bevorzugten Ausführungsform zeigt;

[0044] **Fig. 8** ein Taktfolgediagramm, das ein Beispiel für den Datenblockübertragungstakt bei der bevorzugten Ausführungsform zeigt;

[0045] **Fig. 9** ein Taktfolgediagramm, das ein Beispiel für den Datenblockübertragungstakt bei der bevorzugten Ausführungsform zeigt;

[0046] **Fig. 10** ein Taktfolgediagramm, das ein Beispiel für den Datenblockübertragungstakt bei der bevorzugten Ausführungsform zeigt;

[0047] **Fig. 11** ein Ablaufdiagramm des von den Targets bei der bevorzugten Ausführungsform ausgeführten Unterprogramms zur Bearbeitung der Unterbrechung der seriellen Übertragung;

[0048] **Fig. 12** ein Ablaufdiagramm des von der Steuerung bei der bevorzugten Ausführungsform ausgeführten Unterprogramms zur Bearbeitung der Taktunterbrechung und

[0049] **Fig. 13** ein Takttdiagramm serieller Übertragungen mit dem Start-Zustand.

#### Beschreibung bevorzugter Ausführungsbeispiele

[0050] Nachstehend werden, beginnend mit der Beschreibung der Systemkonfiguration, die bevorzugten Ausführungsbeispiele des erfindungsgemäßen Datenübertragungsverfahrens beschrieben.

[0051] **Fig. 1** ist ein Blockschaltbild eines Datenübertragungssystems, welches das erfindungsgemäße Datenübertragungsverfahren in seiner bevorzugten Ausführungsform verwendet. Wie in **Fig. 1** gezeigt, enthält dieses System die Mikrocomputer 10, 11 und 12, die über einen seriellen Bus mit drei Signalleitungen, nämlich einer Datenleitung 1, einer Taktleitung 2 und einer /BSY-Leitung 3, miteinander verbunden sind.

[0052] Die Datenleitung 1 ist eine Leitung für positiv logische Signale zur Übermittlung serieller Daten, die sowohl mit dem Dateneingangsport 16 als auch mit dem Datenausgangsport 17 der verschiedenen Mikrocomputer verbunden ist. Die Datenausgangsport 17 sind offene Drains, so dass mehrere Mikrocomputer die Signalleitungen gleichzeitig ansteuern können.

- [0053] Der Mikrocomputer 10 gibt den Ein/Aus-Takt der seriellen Daten vom Taktausgangsport 18 an die Taktleitung 2 aus. Die anderen Mikrocomputer lesen diesen Takt über die jeweiligen Takteingangsport 15.
- [0054] Die /BSY-Leitung 3 ist die Signalleitung für das /BSY-Signal. Wenn die /BSY-Leitung 3 auf HIGH ist, sendet keiner der Mikrocomputer (keine der Einrichtungen) am Bus Datenblöcke. Wenn die /BSY-Leitung 3 auf LOW ist, erfolgt die Übermittlung eines Datenblocks oder eines Nullblocks zwischen den Mikrocomputern. Das /BSY-Signal ist ein negatives logisches Signal. Die Mikrocomputer 10, 11 und 12 geben das /BSY-Signal über den /BSY-Signal-Ein/Aus-Port 14 aus und prüfen, ob das /BSY-Signal von einem der anderen Mikrocomputer ausgegeben wird. Das Aufprägen des L-Pegels auf die /BSY-Leitung 3 ist gleichbedeutend mit der Ausgabe des /BSY-Signals. Die Unterbrechung der Aufprägung des L-Signals auf die /BSY-Leitung 3 ist gleichbedeutend mit der Unterbrechung des /BSY-Signals.
- [0055] In der nachstehenden Beschreibung wird unter der Ausgabe des L-Pegels an die /BSY-Leitung 3 die „Aufprägung des /BSY-Signals“ und unter der Beendigung der Ausgabe des L-Pegels an die /BSY-Leitung 3 die „Negierung des /BSY-Signals“ verstanden. Die /BSY-Leitung 3 ist mit jedem Mikrocomputer über eine verdrahtete ODER-Schaltung verbunden, so dass die /BSY-Leitung 3 L-Pegel haben muss, wenn einer der Mikrocomputer das /BSY-Signal aufprägt.
- [0056] Viele Ein-Chip-Mikrocomputer ermöglichen eine serielle 8-Bit-Datenübertragung in einer Hardwarekonstruktion, die nachstehend als „Schiebepuffer“ bezeichnet wird. Der Schiebepuffer umfasst zwei Schieberegister (einen Sendepuffer und einen Empfangspuffer) sowie periphere Schaltungen. Aus einer externen Quelle stammender Takt (externer Takt) und im Mikrocomputer erzeugter Takt (interner Takt) werden zur Steuerung von Datenein- und -ausgang verwendet. Es sei darauf hingewiesen, dass einige Mikrocomputer ein einziges, gemeinsames Schieberegister sowohl als Sende- als auch als Empfangspuffer verwenden. Bei der seriellen Datenübertragung werden Daten aus dem Sendepuffer ab dem höchstwertigen Bit (MSB) bei Taktanfallflanke gesendet und serielle Daten ab dem MSB an der Taktanstiegsflanke in den Empfangspuffer eingespeist.
- [0057] Der Ausgangsanschluss der vom Sendepuffer gesendeten seriellen Daten ist der Datenausgangsport 17, und der Eingangsanschluss für die in den Empfangspuffer eingespeisten seriellen Daten ist der Dateneingangsport 16. Als Eingang und Ausgang wird ein einzelner Ein/Aus-Taktport verwendet. Dieser Rest dient als Ausgangsport, wenn der Schiebepuffer Daten nach dem internen Takt übermittelt, und als Eingangsport, wenn der Schiebepuffer Daten nach dem externen Takt übermittelt. Der Mikrocomputer 10 muss den Takt ausgeben und arbeitet daher nach dem internen Takt, die anderen Mikrocomputer jedoch arbeiten nach dem externen Takt. Der Takt-Ein/Aus-Port dient daher als Taktausgangsport 18 für den Mikrocomputer 10, jedoch als Takteingangsport 15 für die anderen Mikrocomputer.
- [0058] Die in der nachfolgenden Beschreibung des Ausführungsbeispiels verwendete Terminologie hat nachstehende Bedeutung:
- [0059] Steuereinheit bzw. Steuerung: Von den verschiedenen an den seriellen Bus angeschlossenen Mikrocomputern ist der Mikrocomputer, der den Takt ausgibt, die Steuereinheit.
- [0060] Das System hat nur eine Steuereinheit. Der Mikrocomputer 10 ist die Steuereinheit für das in Fig. 1 gezeigte System und wird im folgenden als „Steuereinheit“ bzw. Steuerung bezeichnet.
- [0061] Targets: Von den verschiedenen an den seriellen Bus angeschlossenen Mikrocomputern sind alle außer der Steuereinheit Targets.
- [0062] Mindestens ein Target muss an den seriellen Bus angeschlossen sein. In Fig. 1 sind die Mikrocomputer 11 und 12 Targets. Daher werden die Mikrocomputer 11 und 12 im folgenden als Target 1 bzw. Target 2 bezeichnet.
- [0063] Hauptrechner: Der jeweils Datenblöcke sendende Mikrocomputer ist der Hauptrechner. Die Steuereinheit und alle Targets (Targets 1 und 2 in Fig. 2) können Hauptrechner werden, jedoch können nicht mehrere Mikrocomputer gleichzeitig Hauptrechner sein, d. h. zu einem gegebenen Zeitpunkt kann es immer nur einen Hauptrechner geben.
- [0064] Nebenrechner: Der den Datenblock empfangende Mikrocomputer ist der Nebenrechner. Die Datenübermittlung erfolgt vom Hauptrechner zum Nebenrechner.
- [0065] Auf die Funktion Nebenrechner beschränkte Mikrocomputer, d. h. Mikrocomputer, die keine Datenblöcke senden, können ebenfalls an den seriellen Bus angeschlossen werden.
- [0066] Datenframe bzw. -block: Ein von einem der Mikrocomputer gesendeter kontinuierlicher Datenstrom ist ein Datenblock. Jeder Datenblock enthält einen 1-Byte-Datenkopf und Userdaten.
- [0067] Bytetakt: Der zur Übertragung von einem Byte erforderliche Takt ist der Bytetakt.
- [0068] Übertragungseinheit: Kleinstmögliche Datenübertragungseinheit; in einer Übertragungseinheit können mehrere Datenblöcke übertragen werden.
- [0069] Jede Übertragungseinheit umfasst eine L- und eine H-Phase des /BSY-Signals. Datenblöcke werden während der LOW-/BSY-Signalphase übertragen; das CHK-Byte und andere Steuerdaten werden während der HIGH-/BSY-Signalphase übertragen.
- [0070] Die prinzipielle zeitliche Abfolge von seriellen Daten-, Takt- und /BSY-Signalen wird im folgenden unter

Bezugnahme auf Fig. 2 beschrieben.

[0071] Fig. 2 ist ein Taktfolgediagramm, welches die Beziehung zwischen seriellen Daten, Taktfrequenzen und der /BSY-Leitung bei dem bevorzugten Ausführungsbeispiel zeigt. Wie in Fig. 2a) gezeigt, werden über die Datenleitung 1 gesendete serielle Daten jeweils bitweise bei der Taktabfallflanke ausgegeben. Die seriellen Daten werden bei der Taktanstiegsflanke gelesen (eingegeben). Der Taktzyklus ist  $T_a$ , und Übertragungen erfolgen in Abständen jeweils in 8-Bit-Blöcken.

[0072] Als Bytetakt sind die acht zur Übermittlung von 8 Datenbits erforderlichen Takte festgelegt.

[0073] Wie in Fig. 2b) gezeigt, wird der Bytetakt intermittierend im Bytetaktzyklus  $T_b$  ausgegeben. Das /BSY-Signal wird zu Beginn einer Übertragungseinheit von der Steuereinheit aufgeprägt. Unmittelbar nach Aufprägung des /BSY-Signals gibt die Steuereinheit den Bytetakt aus. Während der Aufprägung des /BSY-Signals werden ein oder mehrere Bytetakte von der Steuereinheit ausgegeben. Wenn das /BSY-Signal auf LOW ist, kann jeder der Mikrocomputer einen Datenblock senden. Die /BSY-Signal-Verneinung wird ebenfalls von der Steuereinheit vorgenommen, und zwar unmittelbar vor Ausgabe des Bytetakts. Während der H-Phase des /BSY-Signals werden normalerweise zwei Bytetakte ausgegeben.

[0074] Die Übertragungseinheit ist als der Zeitabschnitt zwischen aufeinanderfolgenden Pegelabfällen der /BSY-Leitung 3 definiert. Der Übertragungseinheitszyklus ist  $T_c$ . Anschließend werden die über die Datenleitung 1 gesendeten Daten beschrieben. Die über die Datenleitung 1 sendbaren Daten sind in Tabelle 1 zusammengefasst.

Tabelle 1

/BSY	Bezeichnung		Byte	Inhalt
LOW	K o p f	Null- kopf	1	Besagt, dass kein Datenblock gesendet wird
		Daten- kopf	1	Datenart und Anzahl der Bytes
		User- kopf	N	Übertragungsdaten (N = 1-32)
HIGH	D a t e n	CHK- Byte	1	Fehlererfassungscode für Daten, die gesendet werden, wenn das /BSY-Signal LOW ist
		REQ- Byte	1	Datenfehler an Hauptrechner melden ODER Datenausgabeanforderung

[0075] Der Kopf ist ein Datenbyte, das gesendet wird, wenn das /BSY-Signal LOW ist. Das Kopfformat ist in Fig. 3 wiedergegeben; es enthält ein Paritätsbit und eine Sieben-Bit-Sendekennung. Das höchstwertige Bit (Most Significant Bit, MSB) ist das geradzahlige Paritätsbit der Sendekennung. Der Sendekennungswert kann auch im Bereich von 0 bis 127 liegen. Der Kopf ist als einer von drei Typen (Tabelle 2) durch den Wert der Sendekennung bezeichnet.

Tabelle 2

Sendekennung (Bits 6-0)	Kopfbezeichnung
0 (0000000B)	(reserviert)
1-126 (0000001B-1111110B)	Datenkopf
127 (1111111B)	Nullkopf



[0076] Die Sendekennung 0 ist für eine künftige Erweiterung reserviert und wird derzeit nicht verwendet. Köpfe mit einer Sendekennung von 1–126 sind Datenköpfe. Auf den Datenkopf folgen die Userdaten. Köpfe mit der Sendekennung 127 sind Nullköpfe mit dem Wert FFh (hexadezimal).

[0077] Jeder Datenblock umfasst einen Datenkopf und Userdaten, wie in Fig. 4 gezeigt. Die nach dem Datenkopf gesendeten Userdaten können einen Umfang von 1 bis 32 Bytes haben. Die Userdatenbytesanzahl N wird aus der Sendekennung ermittelt. Die Beziehung zwischen dem Datenkopfwert und der Anzahl von Userdatenbytes N ist in Fig. 5 dargestellt. Wie in Fig. 5 gezeigt, werden die Datenköpfe nach dem Datenkopfwert in die Gruppe A, B oder C eingeordnet. In Fig. 5 kann der Wert von X 0 oder 1 sein. Das 6. Bit des Datenkopfes der Gruppe A ist 0. Der Datenkopf der Gruppe A zeigt die Byteanzahl N-1 mit den niedrigsten fünf Bits. Die Bits 6 und 5 des Datenkopfes von Gruppe B sind 1 bzw. 0. Der Datenkopf der Gruppe B zeigt die Bytezahl N-1 mit den niedrigsten vier Bits. Die Bits 6 und 5 des Datenkopfes von Gruppe B sind 1 bzw. 1. Der Datenkopf der Gruppe C zeigt die Bytezahl N-1 mit den niedrigsten drei Bits.

[0078] Das erfindungsgemäße Datenübertragungsverfahren kann 126 Arten von Datenblöcken übertragen, weil die Sendekennungen des Datenblocks zwischen 1 und 126 liegen können. Die Anzahl der Userdatenbytes ist jedoch beschränkt, wie in Tabelle 3 gezeigt, weil die Anzahl der Userdatenbytes aus der Sendekennung bestimmt werden kann, wie in Fig. 5 gezeigt.

Tabelle 3

Userdatenbytes N	Anzahl verfügbarer Sendekennungen	Einteilung		
		Gruppe A	Gruppe B	Gruppe C
1	7	1	2	4
2 – 7	8	2	2	4
8	7	2	2	3
9 – 16	4	2	2	-
17 - 32	2	2	-	-

[0079] Mit fünf Userdatenbytes können sich beispielsweise bis zu acht Datenblocktypen ergeben. In Tabelle 4 sind Beispielwerte nach Datenkopfgruppe angegeben.

Tabelle 4  
Tabelle 4-1

Gruppe	Anzahl Bytes	Datenkopf	
		hexadezimal	binär
A	1	A0	(1010 0000)
	2	81	(1000 0001)
		21	(0010 0001)
	3	82	(1000 0010)
		22	(0010 0010)
	4	03	(0000 0011)
		A3	(1010 0011)
	5	84	(1000 0100)
		24	(0010 0100)
	6	05	(0000 0101)
		A5	(1010 0101)
	7	06	(0000 0110)
		A6	(1010 0110)
	8	87	(1000 0111)
		27	(0010 0111)
	9	88	(1000 1000)
		28	(0010 1000)
	10	09	(0000 1001)
		A9	(1010 1001)
	11	0A	(0000 1010)
		AA	(1010 1010)
	12	8B	(1000 1011)
		2B	(0010 1011)
	13	0C	(0000 1100)
		AC	(1010 1100)
	14	8D	(1000 1101)
		2D	(0010 1101)
	15	8E	(1000 1110)
		2E	(0010 1110)
	16	0F	(0000 1111)
		AF	(1010 1111)

Tabelle 4-2

Gruppe	Anzahl Bytes	Datenkopf	
		hexadezimal	binär
A	17	90	(1001 0000)
		30	(0011 0000)
	18	11	(0001 0001)
		B1	(1011 0001)
	19	12	(0001 0010)
		B2	(1011 0010)
	20	93	(1001 0011)
		33	(0011 0011)
	21	14	(0001 0100)
		B4	(1011 0100)
	22	95	(1001 0101)
		35	(0011 0101)
	23	96	(1001 0110)
		36	(0011 0110)
	24	17	(0001 0111)
		B7	(1011 0111)
	25	18	(0001 1000)
		B8	(1011 1000)
	26	99	(1001 1001)
		39	(0011 1001)
	27	9A	(1001 1010)
		3A	(0011 1010)
	28	1B	(0001 1011)
		BB	(1011 1011)
	29	9C	(1001 1100)
		3C	(0011 1100)
	30	1D	(0001 1101)
		BD	(1011 1101)
	31	1E	(0001 1110)
		BE	(1011 1110)
	32	9F	(1001 1111)
		3F	(0011 1111)

Tabelle 4-3

Gruppe	Anzahl Bytes	Datenkopf	
		hexadezimal	binär
B	1	C0	(1100 0000)
		50	(0101 0000)
	2	41	(0100 0001)
		D1	(1101 0001)
	3	42	(0100 0010)
		D2	(1101 0010)
	4	C3	(1100 0011)
		53	(0101 0011)
	5	44	(0100 0100)
		D4	(1101 0100)
	6	C5	(1100 0101)
		55	(0101 0101)
	7	C6	(1100 0110)
		56	(0101 0110)
	8	47	(0100 0111)
		D7	(1101 0111)
	9	48	(0100 1000)
		D8	(1101 1000)
	10	C9	(1100 1001)
		59	(0101 1001)
	11	CA	(1100 1010)
		5A	(0101 1010)
	12	4B	(0100 1011)
		DB	(1101 1011)
	13	CC	(1100 1100)
		5C	(0101 1100)
	14	4D	(0100 1101)
		DD	(1101 1101)
	15	4E	(0100 1110)
		DE	(1101 1110)
	16	CF	(1100 1111)
		5F	(0101 1111)

Tabelle 4-4

Gruppe	Anzahl Bytes	Datenkopf	
		hexadezimal	binär
C	1	60	(0110 0000)
		E8	(1110 1000)
		F0	(1111 0000)
		78	(0111 1000)
	2	E1	(1110 0001)
		69	(0110 1001)
		71	(0111 0001)
		F9	(1111 1001)
	3	E2	(1110 0010)
		6A	(0110 1010)
		72	(0111 0010)
		FA	(1111 1010)
	4	63	(0110 0011)
		EB	(1111 1011)
		F3	(1111 0011)
		7B	(0111 1011)
	5	E4	(1110 0100)
		6C	(0110 1100)
		74	(0111 0100)
		FC	(1111 1100)
	6	65	(0110 0101)
		ED	(1110 1101)
		F5	(1111 0101)
		7D	(0111 1101)
	7	66	(0110 0110)
		EE	(1110 1110)
		F6	(1111 0110)
		7E	(0111 1110)
	8	E7	(1110 0111)
		6F	(0110 1111)
		77	(0111 0111)

[0080] Das CHK-Byte (siehe Tabelle 1) ist das Fehlerprüfbyte; es ist das Byte mit der geradzahlgigen Parität für alle während des L-Pegels des /BSY-Signals gesendeten Datenbytes. Das CHK-Byte wird von der Steuereinheit berechnet und übermittelt. Die Steuereinheit führt eine Exklusiv-ODER-Funktion in Byteeinheiten bei allen während der L-Phase des /BSY-Signals gesendeten Daten aus und sendet das Rechenergebnis zum ersten Bytetakt nach dem /BSY-Signal-Anstieg. Für die von der Steuerung gesendeten Daten sind die für diese Rechenoperation verwendeten Daten die Daten vor der Sendung. Für die von den Targets gesendeten Daten werden für die Rechenoperation die über die Datenleitung 1 empfangenen Daten verwendet.

[0081] Das REQ-Byte sind die Daten, die für Datenblocksendeanforderungen verwendet werden. Die obere und die untere Tetrade haben denselben Wert. Dieses Byte wird von allen Mikrocomputern einschließlich der festgelegten Nebenrechner gesendet. Im Unterschied zu den anderen Bytes muss dieses Byte gleichzeitig gesendet und empfangen werden. Das REQ-Byte-Format ist in Fig. 6 dargestellt; die obere und die untere Tetrade haben denselben Wert, wie in Fig. 6 gezeigt.

[0082] Ein /ER-Bitwert von 0 bedeutet, dass in den während des L-Pegels des /BSY-Signals gesendeten Daten ein Übermittlungsfehler aufgetreten ist. Alle Mikrocomputer müssen eine Fehlererfassungsroutine unter Verwendung der während des L-Pegels des /BSY-Signals gesendeten Daten ausführen. Es gibt zwei Fehlererfassungsverfahren, nämlich 1. Erfassung von Fehlern im Datenblock unter Verwendung des CHK-Bytes und 2. Erfassung von Fehlern im Kopf unter Verwendung des Paritätsbits. Das erstgenannte Fehlererfassungsverfahren wird von allen Targets unter Verwendung des von der Steuerung gesendeten CHK-Bytes ausgeführt.

Das zweitgenannte Fehlererfassungsverfahren wird von allen Mikrocomputern ausgeführt.

[0083] Wird ein Fehler festgestellt, sendet der erfassende Mikrocomputer ein REQ-Byte, bei dem der /ER-Bit-Wert 0 ist. Der Mikrocomputer, der den Datenblock der entsprechenden Transfereinheit gesendet hat, bestätigt, dass die Daten korrekt empfangen wurden, wenn er einen /ER-Bit-Wert von 1 erfasst. Ob bei Feststellung eines Datenfehlers die Daten erneut gesendet werden, wird von den einzelnen Mikrocomputern festgelegt. Wenn beispielsweise der gesendete Wert sich bis zu dem Zeitpunkt, zu dem er erneut gesendet werden könnte, bereits geändert hat, was bei Zeitdaten der Fall ist, sollte statt der erneuten Sendung des vorherigen Werts eher der aktuelle Wert gesendet werden. Betriebszustandsdaten des Mikrocomputers und sonstige wichtige Daten müssen jedoch erneut gesendet werden.

[0084] Ein /NDR-Bit-Wert von 0 zeigt den anderen Mikrocomputern an, dass einer der an den seriellen Bus angeschlossenen Mikrocomputer eine Datenblock-Sendeanforderung ausgegeben hat. Bei dieser Ausführungsform des Datenübertragungsverfahrens wird ein Datenblock nur dann gesendet, wenn sich ein im Datenblock enthaltener Userdatenwert geändert hat. Es wird daher von einer „ereignisgesteuerten Datenübertragung“ gesprochen, weil ein Datenblock nur dann übertragen wird, wenn ein Ereignis eingetreten ist, das einen Datenblockwert ändert.

[0085] Bei den ereignisgesteuerten Datenübertragungssystemen gehen jedoch die in einem Mikrocomputer gespeicherten Empfangsdaten verloren, wenn der betreffende Mikrocomputer zurückgesetzt wird. Bei der ereignisgesteuerten Datenübertragung jedoch senden der oder die Mikrocomputer, die die verlorengegangenen Daten gesendet haben, diese Daten nur dann erneut, wenn eine Änderung bei einem Datenwert eingetreten ist. Das bedeutet, dass der zurückgesetzte Mikrocomputer die angeforderten Daten nicht erhalten kann und die Funktion des zurückgesetzten Mikrocomputers nicht gewährleistet ist. Das /NDR-Bit wird dazu verwendet, solche Situationen zu vermeiden. Wenn der von der Steuerung und vom Target zu empfangende Datenblock nicht empfangen wird, wird ein REQ-Byte mit einem /NDR-Bit-Wert von 0 gesendet. Wenn das /NDR-Bit des empfangenen REQ-Byte 0 ist, sendet jeder Mikrocomputer alle Datenblöcke, die er senden sollte, bis ein REQ-Byte mit einem /NDR-Bit-Wert von 1 empfangen wird.

[0086] Im folgenden wird beschrieben, in welcher Reihenfolge die Datenblöcke gesendet werden.

[0087] Die Sendereihenfolge der Datenblöcke und der Nullblöcke ist für jeden Mikrocomputer so festgelegt; wie in Tabelle 5 angegeben.

Tabelle 5

Sendereihenfolge		Mikrocomputer
zuerst	0	Steuerung (Mikrocomputer 10)
	1	Target 1 (Mikrocomputer 11)
	2	Target 2 (Mikrocomputer 12)
	...	...
zuletzt	K	Target K

[0088] Das Senden von Datenblöcken beginnt mit dem in der Sendereihenfolge an erster Stelle stehenden Mikrocomputer, nachdem das /BSY-Signal den L-Pegel angenommen hat. Mikrocomputer, die keinen Datenblock zu senden haben, zeigen an, dass kein Datenblock zu senden ist, indem sie einen Nullblock von 1 Byte senden. Der Nullblock enthält einen Nullkopf von 1 Byte, der den Wert FFh (hexadezimal) hat, wie bereits erwähnt. Die Datenleitung 1 ist eine Leitung für positiv logische Signale, die hochgezogen wird. Das Nichtsenden von Daten ist folglich gleichbedeutend mit dem Senden eines Nullkopfs.

[0089] Jeder Mikrocomputer kann den Datenblock mit dem Takt des nächsten Byte senden, wenn alle an früherer Stelle in der Sendereihenfolge stehenden Mikrocomputer entweder das Blocksenden beendet oder angezeigt haben, dass kein Block zu senden ist (d. h. einen Nullblock gesendet haben). Mit anderen Worten kann der Mikrocomputer K in der Sendereihenfolge mit dem Datenblocksenden im Takt des nächsten Byte beginnen, wenn alle von den anderen Mikrocomputern gesendeten Nullblöcke und Datenblöcke gleich K sind, nachdem die /BSY-Leitung 3 den L-Pegel angenommen hat.

[0090] Im folgenden wird die zeitliche Abfolge des Aufprägens und des Negierens des /BSY-Signals beschrieben.

[0091] Die Steuerung sendet Daten- oder Nullblöcke, nachdem sie die /BSY-Leitung 3 von H-Pegel auf L-Pe-

gel gebracht hat, indem sie das /BSY-Signal zu Beginn der Übertragungseinheit aufgeprägt hat. Das Target, das einen Datenblock senden möchte, prägt das /BSY-Signal nach Bestätigung, dass die /BSY-Leitung 3 auf LOW ist, auf.

[0092] Wenn ein Target das Senden von Datenblöcken beendet hat, negiert es das /BSY-Signal unmittelbar nach Sendung der letzten Daten. Dabei bleibt die /BSY-Leitung 3 auf L-Pegel, weil die Steuerung das /BSY-Signal aufprägt.

[0093] Nachdem die Steuerung das Datenblock- oder Nullblocksenden beendet hat, negiert sie zunächst das /BSY-Signal, bevor sie den Bytetakt oder das CHK-Byte sendet. Wenn die /BSY-Leitung 3 auf LOW bleibt, obwohl das /BSY-Signal negiert wird, gibt die Steuerung ein LOW an die /BSY-Leitung 3 aus, bevor sie den Bytetakt ausgibt. Wenn die /BSY-Leitung 3 durch die Negierung des /BSY-Signals den H-Pegel annimmt, wird bei negiertem /BSY-Signal das CHK-Byte gesendet.

[0094] Im folgenden wird unter Bezugnahme auf Fig. 7 bis 10 der zeitliche Ablauf des Datenblocksendens beschrieben.

[0095] Fig. 7 bis 10 zeigen den zeitlichen Ablauf des Datenblocksendens bei einem System, bei dem die Steuerung mit zwei Targets verbunden ist. Fig. 7 ist ein Zeitplan der Transfereinheiten, wenn alle Mikrocomputer Datenblöcke senden. Fig. 8 ist ein Zeitplan der Transfereinheiten, wenn die zwei Mikrocomputer von Steuerung und Target 2 Datenblöcke senden. Fig. 9 ist ein Zeitplan der Transfereinheit, wenn nur das Target 1 einen Datenblock sendet. Fig. 10 ist ein Zeitplan der Transfereinheit, wenn von keinem der Mikrocomputer Daten gesendet werden.

[0096] Um die nachfolgende Beschreibung unter Bezugnahme auf Fig. 7 bis 10 zu vereinfachen, soll die Anzahl der im Datenblock enthaltenen Userdatenbytes zwei sein. In diesen Zeichnungen werden auch die nachstehenden Bezugszeichen verwendet: 71 für die Bytetaktausgabe durch die Steuerung an die Taktleitung 2; 73, 75 und 77 bezeichnen den Datenausgang an die Datenleitung 1 durch die Steuerung bzw. das Target 1 und das Target 2. Diese Daten werden in die Datenleitung 1 als verdrahtetes ODER-Signal eingespeist. Die von der Steuerung bzw. dem Target 1 bzw. dem Target 2 an die /BSY-Leitung 3 ausgegebenen /BSY-Signale sind mit 74, 76, 78 bezeichnet. Diese Signale sind ebenfalls verdrahtetes ODER an die /BSY-Leitung 3, was zu dem Signalpegel 72 auf der /BSY-Leitung 3 führt.

[0097] Gemäß Fig. 7 sendet die Steuerung den Datenkopf nach Aufprägung des /BSY-Signals zu Beginn der Übertragungseinheit. Weil bei den Targets 1 und 2 eine Datensendeanforderung vorliegt, prägen sie sofort nach Empfang dieses Datenkopfes (Zeit  $t_1$ ) das /BSY-Signal auf. Die Steuerung sendet dann die 2-Byte-Userdaten, beendet das Datenblocksenden und negiert das /BSY-Signal zum Zeitpunkt  $t_2$ .

[0098] Die /BSY-Leitung 3 bleibt auf LOW, obwohl die Steuerung das /BSY-Signal negiert hat, weil die Targets 1 und 2 dabei das /BSY-Signal aufprägen. Infolgedessen prägt die Steuerung das /BSY-Signal erneut auf und gibt dann den Bytetakt aus. Das Target 1 kann jetzt mit den Datenblocksenden nach diesem Bytetakt beginnen, weil die Steuerung als einziges vorrangiges Element in der Sendefolge das Datenblocksenden beendet hat. Nachdem das Target 1 das Datenblocksenden beendet hat, negiert es das /BSY-Signal zum Zeitpunkt  $t_3$ . Dabei bleibt die /BSY-Leitung 3 auf LOW, weil das Target 2 und die Steuerung das /BSY-Signal aufprägen.

[0099] Zum Zeitpunkt  $t_3$  haben das Target 1 und die Steuerung, die beide in der Sendefolge gegenüber dem Target 2 vorrangig sind, das Datenblocksenden beendet, so dass das Target 2 mit dem Datenblocksenden nach dem nächsten Bytetakt beginnt. Wenn das Datenblocksenden zum Zeitpunkt  $t_5$  beendet ist, prägt das Target 2 das /BSY-Signal auf. Zum Zeitpunkt  $t_6$  negiert die Steuerung das /BSY-Signal und sendet das CHK-Byte nach Bestätigung, dass die /BSY-Leitung 3 H-Pegel hat.

[0100] Wie oben beschrieben, ist das CHK-Byte das Rechenergebnis einer Exklusiv-ODER-Funktion, die in Byteeinheiten an allen Daten ausgeführt wird, die während des L-Pegels der /BSY-Leitung 3 über die Datenleitung 1 gesendet wurden. Die Targets 1 und 2 empfangen dieses CHK-Byte und verwenden es zur Erfassung von Fehlern in den gesendeten Daten. Beim nächsten Bytetakt senden alle Mikrocomputer das REQ-Byte. Das REQ-Byte ist ebenfalls ein verdrahtetes ODER auf der Datenleitung 1, und das Ergebnis wird in jeden der Mikrocomputer eingespeist.

[0101] Selbst wenn weitere Mikrocomputer an den seriellen Bus angeschlossen werden und ein zusätzlicher Mikrocomputer einen Datenblock sendet, ist das Datenblocksenden ohne Kollision mit den von der Steuerung und den Targets 1 und 2 ausgegebenen Daten möglich, wenn dem zusätzlichen Mikrocomputer eine höhere Sendefolgennummer zugeordnet wird als dem Target 2. Die Steuerung prüft stets, ob von einem der anderen Mikrocomputer ein /BSY-Signal ausgegeben wird, bevor sie den Bytetakt ausgibt, wenn ein Target einen Datenblock sendet. Das dient der Prüfung, ob das /BSY-Signal während des Datenblocksendens infolge einer Targetrücksetzung oder eines sonstigen anomalen Zustands negiert worden ist, und erlaubt somit die sofortige Beendigung der Übertragungseinheit, wenn ein Problem auftaucht.

[0102] Nach dem in Fig. 8 dargestellten Zeitablaufplan erfolgt das Datenblocksenden nur durch die Steuerung und das Target 2. Dabei ist die Steuerungsfunktion die gleiche wie beziehungsweise auf Fig. 7 beschrieben und wird daher im folgenden nicht mehr erwähnt. Das Target 1 jedoch prägt das /BSY-Signal selbst dann nicht auf, wenn für die /BSY-Leitung 3 ein L-Pegel erfasst wird (Zeitpunkt  $t_{11}$ ), weil es keine Datenblöcke sendet.

Die Steuerung negiert das /BSY-Signal, nachdem sie das Datenblocksenden beendet hat (Zeitpunkt t12), prägt jedoch anschließend das /BSY-Signal erneut auf und gibt den Bytetakt aus, weil die /BSY-Leitung 3 auf LOW bleibt. Das Target 1 sendet mit diesem Bytetakt einen Nullkopf, weil es keine Daten zu senden hat.

[0103] In der Praxis wird selbst dann, wenn das Target 1 nichts sendet, ein Nullkopf effektiv gesendet, weil der Nullkopfwert, wie oben beschrieben, FFh (hexadezimal) ist. Das Target 2 erfasst das Datenblock- und Nullblocksenden durch die Steuerung und das Target 1 und beginnt dann mit dem Datenblocksenden. Die anschließende Operation ist die gleiche wie oben bei Fig. 7 beschrieben.

[0104] Fig. 0 ist ein Übertragungseinheit-Zeitablaufplan, wenn nur das Target 1 Datenblöcke sendet. In diesem Fall prägt die Steuerung das /BSY-Signal auf und sendet dann einen Nullkopf. Nach Beendigung des Nullkopfsendens gibt die Steuerung den Bytetakt aus, weil die /BSY-Leitung 3 auf LOW bleibt, selbst wenn das /BSY-Signal negiert wird (Zeitpunkt t21). Das Target 1 erfasst das Nullkopfsenden durch die Steuerung und beginnt mit dem Datenblocksenden synchron mit diesem Bytetakt. Wie in Fig. 9 gezeigt, startet die Steuerung die Transfereinheit, selbst wenn die Steuerung selbst keinen Datenblock sendet, und prüft, ob die Targets eine Datenblocksendeanforderung ausgegeben haben.

[0105] Fig. 10 ist der Zeitablaufplan für den Fall, dass von keinem der Mikrocomputer am seriellen Bus Datenblöcke gesendet werden. Zum Zeitpunkt t31 prägt die Steuerung das /BSY-Signal auf und sendet dann einen Nullkopf. Die Targets 1 und 2 prägen das /BSY-Signal selbst dann nicht auf, wenn ein Abfall auf der /BSY-Leitung 3 erfasst wird, weil sie keine Datenblocksendeanforderung auszugeben haben. Infolgedessen kommt die /BSY-Leitung 3 auf H-Pegel, wenn die Steuerung das /BSY-Signal zum Zeitpunkt t32 negiert. In der Praxis sendet die Steuerung das CHK-Byte zum Zeitpunkt t32, und von allen Mikrocomputern wird zum Zeitpunkt t33 das REQ-Byte ausgegeben, obwohl während dieser Übertragungseinheit kein Datenblocksenden erfolgt.

[0106] Wie in Fig. 7 bis 10 gezeigt, ändert sich die Anzahl der während des L-Pegels der /BSY-Leitung 3 über die Datenleitung 1 gesendeten Datenbytes mit der Übertragungseinheit, weil die Mikrocomputer am Bus nur dann Datenblöcke senden, wenn eine Datenblocksendeanforderung vorliegt.

[0107] Das oben beschriebene, für das Datenblocksenden durch die Steuerung und die Targets verwendete Unterbrechungsverfahren wird nachstehend ausführlicher behandelt.

[0108] Fig. 11 ist das grundlegende Ablaufdiagramm des von den Targets ausgeführten Unterprogramms serielle Sendeunterbrechungsprozedur (nachstehend „serielle Unterbrechungsprozedur“ genannt). Wenn ein Target ein Byte sendet oder empfängt, erzeugt es eine serielle Übertragungsunterbrechung (im folgenden „serielle Unterbrechung“ genannt) nach dem in Fig. 11 dargestellten Ablauf. Genauer gesagt, wird die in Fig. 11 dargestellte serielle Unterbrechungsprozedur sofort nach Einspeisung des Bytetakts ausgeführt.

[0109] Wenn das Target nach einer Rücksetzung erstmals wieder gebootet wird, wird der INT-Modus auf die Standardeinstellung „Reset“ gestellt. Der INT-Modus ist der im internen Speicher des Mikrocomputers gespeicherte Status. Bei einer seriellen Unterbrechung wird dieser Status geprüft, um festzustellen, welche Prozedur das Unterprogramm serielle Unterbrechungsprozedur ausführen muss. Wenn eine serielle Unterbrechung erstmals nach dem Anlaufen eintritt, wird zunächst Schritt 198 ausgeführt, weil der INT-Modus „Reset“ ist, und anschließend Schritt 199 zur Feststellung des Pegels auf der /BSY-Leitung 3. Ist die /BSY-Leitung 3 auf LOW, wurde der Mikrocomputer mitten in einer Transfereinheit neu gestartet. In diesem Fall muss das Target die Transfereinheit ungültig machen, weil der betreffende gesendete Datenblock möglicherweise von dem Target empfangen werden sollte. Folglich wird das /ER-Bit des RFQ-Byte gelöscht (Schritt 215) und der INT-Modus auf „Fehler“ geändert (Schritt 216) und die serielle Unterbrechungsprozedur endet. Während die /BSY-Leitung 3 danach auf L-Pegel gehalten wird, wird das Unterbrechungsprozedurprogramm beendet, indem einfach der Zustand in den Schritten 198, 200, 218, 219 und 223 bewertet wird. Die Transfereinheit kann ungültig gemacht werden, indem das /ER-Bit des REQ-Registers in der später noch beschriebenen Weise gelöscht wird.

[0110] Wenn die /BSY-Leitung 3 beim Anlaufen auf HIGH ist, wird der INT-Modus in Schritt 235 auf „NOP“ geändert und bleibt „NOP“, bis die /BSY-Leitung 3 den L-Pegel annimmt.

[0111] Wenn die Steuerung zu Beginn der Transfereinheit die /BSY-Leitung 3 auf LOW setzt und einen Kopf sendet, führt das Target bei Kopfeingangsunterbrechung den in Fig. 11 dargestellten Ablauf aus. In Schritt 200 ist für den Signalpegel auf der /BSY-Leitung 3 ein Abfall festgestellt, es werden die Schritte 201 und 202 ausgeführt, wenn eine Sende- oder Empfangsanforderung vorliegt, und das /BSY-Signal wird aufgeprägt. Dann wird unter Verwendung des empfangenen Kopfes und des CHK-Registers (in den Abbildungen nicht eingezeichnet), welches das Register für Paritätsoperationen ist, eine Exklusiv-ODER-Operation ausgeführt (Schritt 203). Das Ergebnis dieser Operation wird im CHK-Register gespeichert.

[0112] Das CHK-Register wird gelöscht (ebenfalls nicht dargestellt), wenn auf der /BSY-Leitung 3 ein Abfall festgestellt wird. Dann wird in den Schritten 204, 205 und 206 festgestellt, ob der empfangene Kopf ein Nullkopf oder der Datenkopf des zu empfangenen Datenblocks ist, oder ob ein Kopffehler vorgekommen ist.

[0113] Wurde ein Nullkopf empfangen, bestimmt der Mikrocomputer, ob er mit dem Datenblocksenden beginnen kann (Schritt 207). Bei dieser Bewertung wird geprüft, ob zwei Bedingungen erfüllt sind, nämlich, ob 1. der betreffende Mikrocomputer das /BSY-Signal aufprägt, und ob 2. alle in der Sendefolge vorrangigen Mikrocom-



puter das Datenblock- oder Nullblocksenden beendet haben. Sind beide Bedingungen erfüllt, wird der INT-Modus in Schritt 208 in „Senden“ geändert, eine Exklusiv-ODER-Operation ausgeführt unter Verwendung der zu sendenden Daten und des Werts im CHK-Register (Schritt 209) und das Ergebnis in das CHK-Register geschrieben. In Schritt 210 werden die zu sendenden Daten auch in den Sendepuffer (Schiebepuffer) eingetragen. Die in den Sendepuffer eingetragenen Daten sind die nach Vorgabe des Bytetakts durch die Steuerung an das Target an die Datenleitung 1 ausgegebenen Daten.

[0114] Wenn der Datenkopf des zu empfangenden Datenblocks empfangen worden ist, wird er in einem Speicherbereich des Mikrocomputers gespeichert (Schritt 211). Außerdem werden der INT-Modus auf „Empfangen“ geändert (Schritt 212), die Anzahl der Bytes der Userdaten aus dem empfangenen Datenkopf entnommen und die Anzahl der Bytes in den Empfangszähler eingegeben.

[0115] In Schritt 206 wird eine Kopparitätsprüfung vorgenommen. Wird im empfangenen Kopf ein Fehler entdeckt, wird das /BSY-Signal negiert (Schritt 214), das /ER-Bit des REQ-Registers (in den Zeichnungen nicht eingezeichnet) gelöscht (Schritt 215), der INT-Modus in „Fehler“ geändert (Schritt 216), und die serielle Unterbrechungsprozedur endet. Wird in dem empfangenen Kopf kein Paritätsfehler entdeckt, wird der INT-Modus in „Kopf warten“ geändert (Schritt 217) und der Empfangszähler eingestellt (Schritt 213).

[0116] Wenn die /BSY-Leitung 3 bei der zweiten oder einer späteren seriellen Unterbrechung nach Beginn der Übertragungseinheit L-Pegel hat, wird der Schritt 218 ausgeführt. Ist der INT-Modus „Kopf erfassen“, sind die im Empfangspuffer empfangenen Daten ein Datenkopf oder ein Nullkopf. Das Unterprogramm springt daher auf Schritt 203 zur Verarbeitung des Kopfeingangs. Ist der INT-Modus „Senden“, sendet der das Unterprogramm ausführende Mikrocomputer den Datenblock. Daher wird geprüft, ob das Senden beendet ist, d. h., ob das letzte Byte des Datenblocks gesendet worden ist (Schritt 220). Ist das Senden nicht beendet, werden die Schritte 209 und 210 ausgeführt, um das Senden eines Datenbytes beim nächsten Bytetakt vorzubereiten. Ist das Senden beendet, wird das /BSY-Signal negiert (Schritt 221).

[0117] Wenn die /BSY-Leitung 3 nach Sendeende zum nächsten Bytetaktausgabezeitpunkt auf L-Pegel bleibt, sendet ein anderer Mikrocomputer einen Kopf und daher ändert sich der INT-Modus auf „Kopf erfassen“ (Schritt 222). Ist der INT-Modus „Error“ („Fehler“), wird das Unterprogramm Unterbrechungsverarbeitung beendet, ohne dass etwas geschieht (Schritt 223). Ist der INT-Modus „Empfangen“ oder „Kopf warten“, wird an den empfangenen Daten und an den CHK-Register-Daten eine Exklusiv-ODER-Operation vorgenommen (Schritt 224) und das Ergebnis in das CHK-Register eingegeben. Ist der INT-Modus „Empfangen“, werden die im Empfangspuffer empfangenen Daten in einem Speicherbereich des Mikrocomputers gespeichert (Schritt 226). Ferner wird der Empfangszähler zurückgestellt (Schritt 227), und in Schritt 228 wird geprüft, ob er auf null steht. Steht der Empfangszähler auf 0, ist das Datenblocksenden durch einen anderen Mikrocomputer abgeschlossen. Daher springt das Unterprogramm auf Schritt 207, in dem festgestellt wird, ob das Senden beginnen kann.

[0118] Wenn für die /BSY-Leitung 3 ein H-Pegel erfasst wird, wenn eine serielle Unterbrechung eintritt, wird das CHK-Byte im Empfangspuffer empfangen. Das empfangene CHK-Byte wird dann mit dem Wert im CHK-Register verglichen (Schritt 229). Stimmen die Werte überein, liegt kein Sendefehler vor, und das Unterprogramm geht zu Schritt 231 über. Stimmen die Werte nicht überein, liegt bei den Daten, die über die Datenleitung 1 übermittelt wurden, während die /BSY-Leitung 3 den L-Pegel hatte, ein Datenfehler vor, weshalb das /ER-Bit des REQ-Bytes zurückgesetzt wird (Schritt 230). Das im REQ-Register gespeicherte REQ-Byte wird dann in den Sendepuffer eingegeben (Schritt 231). Dieses REQ-Byte wird beim nächsten Bytetakt über die Datenleitung 1 gesendet. Außerdem wird der INT-Modus auf „REQ prüfen“ geändert (Schritt 232).

[0119] Kommt es während des INT-Modus „REQ prüfen“ zu einer seriellen Unterbrechung, bleibt die /BSY-Leitung 3 auf H-Pegel, und das REQ-Byte wird im Empfangspuffer empfangen. Dieses REQ-Byte wird im Empfangspuffer als das verdrahtete ODER-Ergebnis auf der Datenleitung 1 der von allen Mikrocomputern ausgegebenen REQ-Bytes empfangen. Infolgedessen wird das /ER-Bit auf 0 gesetzt, wenn einer der Mikrocomputer einen Fehler feststellt.

[0120] Wenn der INT-Modus „REQ prüfen“ in Schritt 233 bestätigt wird, wird eine Senden/Empfangen-Meldung an das Anwendungsprogramm ausgegeben (Schritt 234). Genauer gesagt, wird eine Endemarkierung gesetzt, die anzeigt, dass die Übertragungseinheit beendet ist. Das Anwendungsprogramm weiß, dass die Übertragungseinheit beendet ist, wenn es diese Endemarkierung liest, und liest den empfangenen Datenblock aus dem Speicherbereich, wenn das /ER-Bit des REQ-Bytes 1 ist, weil das /ER-Bit nur dann 1 ist, wenn Senden und Empfangen normal sind. Ist das /ER-Bit 0, übergeht die Anwendung den empfangenen und im Speicherbereich gespeicherten Datenblock. In Schritt 235 wird der INT-Modus dann auf „NOP“ geändert. Der Bytetakt wird normalerweise nur dann zweimal ausgegeben, wenn die /BSY-Leitung 3 auf N-Pegel ist, wenn jedoch die Steuerung wegen Rauschens zurückgesetzt wird, kann sie drei oder mehr Bytetakte ausgeben, wie unten noch beschrieben wird. Der INT-Modus „NOP“ ist für diese Situation vorgesehen.

[0121] Fig. 12 ist ein grundlegendes Ablaufdiagramm des von der Steuerung ausgeführten Unterprogramms „Taktunterbrechungsverarbeitung“. Die Steuerung verwendet Taktunterbrechungen, um den Bytetakt intermittierend auszugeben. Die Steuerung verwendet einen internen Taktgeber, um bei jedem Bytetaktausgabeinter-

vall (Tb) Unterbrechungen zu erzeugen und während der Taktunterbrechungsverarbeitung Datenblöcke in Ein-Byte-Blöcken zu senden oder zu empfangen.

[0122] Zu Beginn der Taktunterbrechungsverarbeitung führt die Steuerung den Schritt 300 aus und führt jeden Vorgang gemäß dem INT-Modus aus. Beim ersten Neustart der Steuerung nach einer Rücksetzung ist der INT-Modus „Reset“ („Rücksetzung“), und die Steuerung prüft daher zunächst den Status der /BSY-Leitung 3 (Schritt 301). Die /BSY-Leitung 3 ist normalerweise auf H-Pegel, wenn das System anläuft, weshalb Leerdaten ausgegeben werden (Schritt 302). Der Schiebepuffer der Steuerung arbeitet nach dem internen Takt, so dass das Datensenden damit beginnt, dass Daten in den Sendepuffer geschrieben werden. Diese Leerdaten sind Daten mit dem Bit an derselben Position wie das auf 0 stehende /ER-Bit. Wenn die /BSY-Leitung 3 nach dem Booten der Steuerung auf H-Pegel ist, werden zwei Bytes Leerdaten in zwei aufeinanderfolgenden Bytes ausgegeben. Wird die Steuerung zurückgesetzt, wenn nur die Steuerung das /BSY-Signal aufprägt, wird das /BSY-Signal negiert und die /BSY-Leitung 3 geht auf H-Pegel. Bei einem Neustart der Steuerung in diesem Zustand, startet die Steuerung während einer Übertragungseinheit. Durch Ausgabe von zwei Bytes Leerdaten zu einer vorgegebenen Zeit, in der die Bitposition des /ER-Bits 0 ist, kann somit die aktuelle Transfereinheit selbst dann unwirksam gemacht werden, wenn die Steuerung während der Transfereinheit zurückgesetzt und das /BSY-Signal negiert wird.

[0123] Wenn die /BSY-Leitung 3 in Schritt 301 auf L-Pegel ist, kann die Steuerung zurückgesetzt worden sein, während ein Target gerade das /BSY-Signal aufgeprägt hat. In diesem Fall wird das /ER-Bit des REQ-Bytes zurückgesetzt (Schritt 305), der INT-Modus auf „Fehler“ geändert (Schritt 306) und mit der Bytetaktausgabe begonnen (Schritt 307). Der Bytetakt wird ausgegeben, indem das Bitmuster FFh (hexadezimal) in den Sendepuffer geschrieben wird, wodurch alle Bits auf der Datenleitung 1 auf HIGH sind. Das ist das Gleiche, wie wenn nur der Bytetakt ohne Daten ausgegeben wird.

[0124] Wenn die zwei Bytes Leerdaten ausgegeben sind, wird der INT-Modus in „Kopf“ geändert (Schritt 304). Daher wird bei der nächsten Taktunterbrechung der Schritt 300 ausgeführt und danach der Schritt 308. Durch Aufprägung des /BSY-Signals (Schritt 308) wird die /BSY-Leitung 3 auf L-Pegel gebracht, und die Übertragungseinheit beginnt. Wenn eine Datenblocksendeanforderung vorliegt, wird der INT-Modus in „Senden“ geändert (Schritt 313), eine Exklusiv-ODER-Operation am CHK-Register-Wert und den zuerst gesendeten Daten, d. h. dem Datenkopf (Schritt 314) ausgeführt und das Ergebnis an das CHK-Register gemeldet. Das CHK-Register wird zurückgesetzt, wenn der INT-Modus in „Kopf“ geändert wird (in den Abbildungen nicht dargestellt). Dann beginnt das Senden eines Bytes der Daten, welche den Datenblock enthalten, indem die zu sendenden Daten in den Sendepuffer eingeschrieben werden (Schritt 315). Der Schritt 315 wird immer dann einmal ausgeführt, wenn anschließend eine Taktunterbrechung vorgenommen wird, um den vollständigen Datenblock Byte für Byte zu übermitteln. Wenn die letzten Daten eines Datenblocks in Schritt 315 in den Sendepuffer eingeschrieben worden sind, wird in Schritt 316 das Sendeende erfasst und der INT-Modus in „Taktausgabe“ geändert (Schritt 312).

[0125] Liegt keine Datenblocksendeanforderung vor (Schritt 309), wird zu Beginn der Übertragungseinheit der Schritt 310 ausgeführt, um einen Nullkopf in das CHK-Register einzufügen. Der Nullkopf wird auch in den Sendepuffer eingeschrieben (Schritt 311), und dann beginnt das Senden des Nullkopfs. Danach wird der INT-Modus in „Takt ausgeben“ geändert (Schritt 312).

[0126] Wenn während des INT-Modus „Taktausgabe“ eine Taktunterbrechung erfolgt, wird der INT-Modus zunächst in „Kopf erfassen“ geändert (Schritt 317). Weil die Steuerung zu diesem Zeitpunkt das Datenblock- oder Nullblocksenden beendet hat, geht die Steuerung zu Schritt 318 über, in dem das /BSY-Signal negiert wird, woraufhin der Pegel der /BSY-Leitung 3 erfasst wird (Schritt 319). Wenn die /BSY-Leitung 3 auf L-Pegel ist, obwohl das /BSY-Signal negiert wird, wird das /BSY-Signal erneut aufgeprägt (Schritt 320), und es beginnt die Ausgabe des Bytetakts. Das in der Sendefolge auf die Steuerung folgende Target sendet mit diesem Bytetakt den Nullkopf oder einen Datenkopf. In diesem Unterbrechungsvorgang erfolgt keine Datenkopfempfangsverarbeitung, sondern erst während des nächsten Taktunterbrechungsvorgangs. Um die Beschreibung des Ablaufs von Schritt 318 bis Schritt 325 zu vereinfachen, wird diese Abfolge als Endeprüfroutine 350 bezeichnet.

[0127] Wenn während des INT-Modus „Kopf erfassen“ eine Taktunterbrechung erfolgt, ist der erste Schritt (331) die Prüfung auf Kopffehler. Wird ein Kopffehler festgestellt, wird das /ER-Bit des REQ-Bytes zurückgesetzt (Schritt 332) und der INT-Modus in „Fehler“ geändert (Schritt 333).

[0128] Wenn der INT-Modus sich in „Fehler“ ändert, wird die Endeprüfroutine 350 wiederholt, bis alle Targets das /BSY-Signal negieren. Wird in Schritt 331 kein Kopffehler festgestellt, wird geprüft, ob der empfangene Kopf ein Nullkopf ist (Schritt 334). Ist der empfangene Kopf ein Nullkopf, wird die Endeprüfroutine 350 erneut ausgeführt um zu prüfen, ob die Targets das /BSY-Signal negiert haben. Wurde ein Nullkopf empfangen, sind die als nächstes von dem Target gesendeten Daten ebenfalls ein Kopf, so dass der INT-Modus im Status „Kopf erfassen“ bleibt. Ist der empfangene Kopf ein Datenkopf, wird der Umfang des vollständigen Datenblocks (d. h. die Userdatenbytegröße + 1) aus dem Datenkopf entnommen und in den Empfangszähler geschrieben (Schritt 335). Ferner wird geprüft, ob der empfangene Datenkopf der Datenkopf eines zu empfangenden Datenblocks ist (Schritt 336). Soll der Datenblock empfangen werden, wird der INT-Modus in „Empfangen“ geän-

- dert (Schritt 337), anderenfalls wird der INT-Modus in „Kopf warten“ geändert (Schritt 338).
- [0129] Ist der INT-Modus „Empfangen“, geht die Steuerung zu Schritt 326 über, und die im Empfangspuffer empfangenen Daten werden im Speicherbereich gespeichert, wenn der INT-Modus jedoch „Kopf warten“ ist, springt das Programm unmittelbar auf Schritt 327.
- [0130] In Schritt 327 wird unter Verwendung der empfangenen Daten eine Paritätsberechnung vorgenommen. In Schritt 328 wird der Empfangszähler zurückgesetzt, und anschließend wird das Ende des Datenblocksendens durch das Target erfasst (Schritt 329). Wenn das Datenblocksenden beendet ist, wird der INT-Modus in „Kopf erfassen“ geändert (Schritt 330) und die Endeprüfroutine 350 ausgeführt. Ist das Senden noch nicht beendet, wird die Endeprüfroutine 350 ausgeführt, ohne den INT-Modus zu ändern.
- [0131] Wenn die /BSY-Leitung 3 in Schritt 319 der Endeprüfroutine 350 den H-Pegel hat, wird geprüft, ob der Empfangszählerwert 0 ist (Schritt 322). Hat der Empfangszähler einen anderen Wert als 0, negiert das Target das /BSY-Signal, ohne die angegebene Anzahl von Userdatenbytes zu senden, und das /ER-Bit wird daher zurückgesetzt (Schritt 323). Ist der Empfangszähler auf 0, wird sofort mit dem CHK-Byte-Senden begonnen, indem der Inhalt des CHK-Registers in den Sendepuffer geschrieben wird. Ist der INT-Modus „Fehler“, wird die Empfangsparitätsberechnung (Schritt 327) nicht ausgeführt, und das CHK-Byte zu Sendebeginn (Schritt 324) weist nicht den korrekten Wert auf. Das ist jedoch unproblematisch, denn wenn der INT-Modus „Fehler“ ist, wird ein REQ-Byte mit zurückgesetztem /ER-Bit gesendet. Der INT-Modus ändert sich dann in „REQ senden“, und die Unterbrechungsprozedur endet.
- [0132] Beginnt die Taktunterbrechungsprozedur im INT-Modus „REQ senden“, wird mit dem Senden des REQ-Byte begonnen, indem das im REQ-Register gespeicherte REQ-Byte in den Sendepuffer geschrieben wird (Schritt 339). Die Taktunterbrechung kann dann beendet werden, indem einfach der INT-Modus auf „REQ verarbeiten“ geändert wird (Schritt 340).
- [0133] Die nächste Taktunterbrechung beginnt im INT-Modus „REQ-Verarbeitung“. Die in Schritt 341 erfolgte Senden/Empfangen-Meldung an das Anwendungsprogramm ist der gleiche Vorgang wie der vom Target in Schritt 234 ausgeführte (Fig. 11). Der INT-Modus wird dann in „NOP“ geändert (Schritt 342), um die Taktunterbrechung zu beenden.
- [0134] Die nächste Taktunterbrechung beginnt im INT-Modus „NOP“, jedoch wird im INT-Modus „NOP“ kein Bytetakt ausgegeben. Genauer gesagt, wird geprüft, ob der aktuelle Takt der Übertragungseinheitsendtakt ist (Schritt 343), und der INT-Modus wird in „Kopf“ geändert (Schritt 345), sobald die Übertragungseinheit nach der nächsten Taktunterbrechung gestartet wird.
- [0135] Bei der vorstehend beschriebenen bevorzugten Ausführungsform der Erfindung muss das Target, das zum Hauptrechner geworden ist, die Daten in den Sendepuffer schreiben, bevor der nächste Bytetakt ausgegeben wird, weil die Steuerung den Bytetakt in gleichbleibenden Abständen ausgibt. Wird der Schreibvorgang nicht rechtzeitig abgeschlossen, erfolgt eine entsprechende Meldung an die Steuerung, und die Ausgabe des Bytetakts wird nach dem nachstehend beschriebenen Verfahren verzögert.
- [0136] Fig. 13 ist ein Zeitablaufplan für serielle Übertragungen mit der in einem Ein-Chip-Mikrocomputer (z. B. MN18888, hergestellt von der Matsushita Electric Industrial Corporation) gegebenen Startbedingung. Es sind der externe Takt und die zeitliche Abfolge für Übertragungen mit bedingtem Start mit Startbedingung angegeben. Wenn die Vorbereitungen zum seriellen Senden von Daten abgeschlossen sind (Zeitpunkt t51), d. h. wenn die zu sendenden Daten in den internen Schiebepuffer geschrieben sind, ändert sich der Status des Datenausgangs von der Impedanz HIGH zu LOW. Wenn in diesem Status der Takteingang beginnt (Zeitpunkt t52), werden beginnend mit dem MSB byteweise nacheinander Daten durch den Datenausgang gesendet. Nach Ausgabe des LSB (Zeitpunkt t53), wird der Datenausgang auf H-Pegel gesetzt. Nach einer vorgegebenen Wartezeit wird der Datenausgang erneut auf H-Pegel gesetzt (Zeitpunkt t54).
- [0137] Die Hauptaufgabe der Startbedingung ist die Synchronisation der Byteeinheiten auf Sende- und Empfangsseite. Vor dem Senden von Daten wird mit anderen Worten die Datenleitung auf L-Pegel gesetzt, während die Taktleitung auf H-Pegel ist, wodurch der Taktzähler auf Empfangsseite zurückgesetzt und die Synchronisation der Byteeinheiten auf Sende- und Empfangsseite ermöglicht wird. Eine Aufgabe der Startbedingung ist es daher, eine Bitverschiebung durch Rauschen zu verhindern. Dank dieser Funktion des bedingten Starts kann die Datenleitung auf L-Pegel gebracht werden, indem das Target einfach die Senddaten in den Schiebepuffer schreibt.
- [0138] Wenn ein Mikrocomputer, der Daten in der in Fig. 13 gezeigten zeitlichen Abfolge seriell übermittelt, als Target verwendet wird, ist der Datenausgang ein Drei-Status-Ausgang, und zwischen den Datenausgang 17 und die Datenleitung 1 ist ein Open-Kollektor-Ausgangsgatter geschaltet.
- [0139] Zwischen den Schritten 320 und 321 der Endeprüfroutine 350 prüft die Steuerung den Status der Datenleitung 1. Ist die Datenleitung auf H-Pegel, wird die Taktunterbrechungsprozedur beendet, ohne den Schritt 321 auszuführen. In diesem Fall muss lediglich in der nächsten Taktunterbrechungsprozedur die Endeprüfroutine 350 ausgeführt werden. Ist die Datenleitung 1 auf L-Pegel, hat das Haupttarget die Vorbereitungen zum Datensenden abgeschlossen, und es beginnt die Ausgabe des Bytetakts.
- [0140] Mit vorstehendem Verfahren kann geprüft werden, ob die Vorbereitung zum Datensenden durch das

die Startbedingung verwendende Target abgeschlossen sind, so dass Datensendeausfälle verhindert werden können.

[0141] Nachfolgend wird ein Verfahren beschrieben, mit dem das Target die Bytetaktausgabe durch die Steuerung erfassen kann, bevor es die Sendevorbereitungen beendet hat.

[0142] Im allgemeinen enthält der Schiebepuffer einen Zähler zum Zählen der durch den Takteingang 15 eingegebenen Takte. Dieser Zähler wird zurückgesetzt, wenn Daten in den Sendepuffer geschrieben werden, und zählt bei Taktanstieg aufwärts. Der Takt wird ebenfalls auf 0 gesetzt, wenn ein 1-Byte-Sendetakt, d. h. acht Takte, eingegeben sind. Infolgedessen ist dieser Zähler immer auf 0, wenn eine serielle Unterbrechung erfolgt. Wenn beispielsweise die Bittaktperiode 10 µsec beträgt, zeigt dieser Zähler 70 µsec lang einen von 0 verschiedenen Wert an. Daher prüft das Target diesen Zähler mindestens alle 70 µsec, während es die serielle Unterbrechungsprozedur ausführt. Zeigt der Zähler einen von 0 verschiedenen Wert an, bevor die Unterbrechungsprozedur abgeschlossen ist, hat die Steuerung den nächsten Bytetakt ausgegeben. In diesem Fall kann das Target die Transfereinheit ungültig machen, indem es das /BSY-Signal negiert, das Senden unterbricht und ein REQ-Byte mit /ER-Bit 0 sendet.

[0143] Anschließend wird die Fehlererfassungsfähigkeit beschrieben. Bei diesem Ausführungsbeispiel wird zur Fehlererfassung eine Paritätsprüfung vorgenommen. Normalerweise können durch Paritätsprüfung keine Fehler erfasst werden, wenn ein Bitfehler an zwei Stellen vorliegt. Wenn beispielsweise in Byteeinheiten Parität hergestellt ist, ist eine Erfassung nicht möglich, wenn in zwei Bits in derselben Position ein Fehler vorliegt. Eine Fehlererfassung ist bei dem vorliegenden Ausführungsbeispiel auch nicht möglich, wenn Fehler in denselben Bitpositionen im selben Datenblock vorliegen, jedoch ist eine Fehlererfassung möglich, wenn Fehler in denselben Bitpositionen in unterschiedlichen Datenblöcken vorliegen. Nehmen wir beispielsweise an, dass die Steuerung und das Target 1 während derselben Transfereinheit den Datenblock A bzw. den Datenblock B senden und dass im Datenblock B an derselben Bitposition wie im Datenblock A ein Bitfehler vorliegt. Jeder Mikrocomputer verwendet die folgenden zwei Arten von Daten nach Berechnung der CHK-Bytes:

- a) Daten vor dem Senden (d. h. korrekte Daten) für die von ihm selbst gesendeten Daten und
- b) Daten, die durch den seriellen Bus von anderen Mikrocomputern eingespeist wurden, für die von anderen Mikrocomputern gesendeten Daten. Folglich verwenden die jeweiligen Mikrocomputer zur CHK-Byte-Berechnung die in Tabelle 6 ausgewiesenen Datenblöcke.

Tabelle 6

	Zur CHK-Byte-Berechnung verwendeter Datenblock
Steuerung	korrekter Datenblock A, Fehlerdatenblock B
Target 1	Fehlerdatenblock A, korrekter Datenblock B
Target 2	Fehlerdatenblock B, Fehlerdatenblock B

[0144] Weil sowohl die Steuerung als auch das Target 1 das CHK-Byte unter Verwendung eines Datenstrangs berechnen, der einen Fehler enthält, sind die Rechenergebnisse dieselben, und es ist keine Fehlererfassung möglich. Das Target 2 berechnet das CHK-Byte jedoch unter Verwendung eines Datenstrangs, der zwei Fehler enthält, so dass das Rechenergebnis nicht mit der CHK-Byte-Ausgabe durch die Steuerung übereinstimmt. Daher sendet das Target 2 ein REQ-Byte mit einem /ER-Bit 0 und meldet auf diese Weise der Steuerung und dem Target 1, dass ein Sendefehler erfasst worden ist. Infolgedessen bietet das erfindungsgemäße Datenübertragungsverfahren eine Fehlererfassungsfähigkeit, die in etwa gleich gut ist wie die Fehlererfassung durch Anfügung eines Ein-Byte-Paritätsprüfbytes an jeden Datenblock, wenngleich ein Ein-Byte-Paritätsprüfbyte für mehrere Datenblöcke verwendet wird.

[0145] Wie vorstehend beschrieben, beginnt die Steuerung bei dem bevorzugten Ausführungsbeispiel des Datenübertragungsverfahrens mit dem Datenblock- oder Nullblocksenden nach Aufprägung des /BSY-Signals auf die /BSY-Leitung 3. Targets mit Datenblocksendeanforderung prägen das /BSY-Signal auf, während die /BSY-Leitung 3 von der Steuerung auf dem L-Zustand gehalten wird. Wenn die /BSY-Leitung 3 im L-Zustand bleibt, obwohl das /BSY Signal negiert wird, gibt die Steuerung den Sendetakt aus, bis die /BSY-Leitung 3 den I-Zustand annimmt. Nachdem die Steuerung und alle an einer früheren Stelle in der Senderangfolge stehenden Targets das Datenblock- oder Nullblocksenden beendet haben, beginnt das in der Sendefolge nächstfolgende Target das Datenblocksenden synchron mit dem Sendetakt und negiert dann das /BSY-Signal, wenn das Datenblocksenden beendet ist. Daher können zusätzliche Mikrocomputer ohne weiteres an den seriellen Bus angeschlossen oder wieder von ihm getrennt werden, und alle Mikrocomputer mit Datenblocksendeanforderung können kollisionsfrei Datenblöcke senden.

[0146] Die Steuerung gibt auch intermittierend den Einheitstakt aus, der zum Senden einer Datenübertra-

gungseinheit erforderlich ist, und zwar nach erneuter Aufprägung des /BSY-Signals, wenn die /BSY-Leitung 3 zum Zeitpunkt der Einheitstaktausgabe auf L-Pegel bleibt, obwohl die Steuerung das /BSY-Signal nach Beendigung des Datenblock- oder Nullblocksendens negiert hat. Die Targets negieren das /BSY-Signal unmittelbar nach Beendigung des Datenblocksendens. Daher ist es nicht nötig, eine externe Unterbrechungsschaltung zur Überwachung der /BSY-Leitung 3 zu verwenden.

[0147] Außerdem berechnet die Steuerung den Fehlererfassungscode für alle während des L-Pegels der /BSY-Leitung 3 über die Datenleitung 1 gesendeten und empfangenen Daten und sendet den Fehlererfassungscode, nachdem die /BSY-Leitung 3 den N-Pegel angenommen hat. Alle Targets erfassen das Vorhandensein von Übertragungsfehlern unter Verwendung von drei Arten von Daten wie folgt:

- a) Daten, die vom jeweiligen Target über die Datenleitung gesendet wurden,
- b) Daten, die über die Datenleitung empfangen wurden,
- c) Fehlererfassungscode, empfangen über die Datenleitung,

und senden bei Erfassung eines Fehlers ein REQ-Byte mit einem /ER-Bit von 0. Daher ist es nicht nötig, jedem Datenblock einen Fehlererfassungscode anzuhängen, und das Datenblocksenden kann effizient erfolgen, weil es nicht nötig ist, jeden Datenblock gesondert zu bestätigen.

[0148] In den Datenkopf wird auch ein Fehlererfassungscode zur Erfassung von Sendefehlern im Datenkopf eingefügt. Wenn ein Target im Datenkopf einen Sendefehler entdeckt, negiert es das /BSY-Signal, um die laufende Übertragungseinheit sofort bei Auftreten eines Fehlers zu beenden, so dass unnötiges Datensenden verhindert wird.

[0149] Zusätzlich senden die Steuerung und alle Targets ein REQ-Byte mit einem /NDR-Bit von 0, wenn ein vom jeweiligen Mikrocomputer zu empfangender Datenblock nicht empfangen worden ist. Wenn das /NDR-Bit des empfangenen REQ-Byte 0 ist, senden die Steuerung und alle Targets nacheinander alle Datenblöcke, die der Sender senden sollte, bis ein REQ-Byte mit einem /NDR-Bit von 1 empfangen wird. Obwohl es sich hier um ein ereignisgesteuertes Datenübertragungsverfahren handelt, können somit alle erforderlichen Datenblöcke rasch wiederhergestellt werden, indem die verlorengegangenen Daten eines anderen Mikrocomputers erneut gesendet werden, indem das /NDR-Bit auf null gesetzt wird, selbst wenn die bis dato empfangenen und gespeicherten Daten durch eine Störung, beispielsweise eine Rücksetzung, verlorengegangen sind.

[0150] Ferner sendet die Steuerung ein REQ-Byte mit einem /ER-Bit von 0, wenn die /BSY-Leitung 3 wieder den H-Pegel annimmt, bevor der Hauptrechner die volle Userdatenlänge sendet, die in dem vom Hauptrechner gesendeten Datenkopf angegeben ist. Dadurch wird es möglich, das Datenblocksenden ungültig zu machen, wenn das den Datenblock sendende Target zurückgesetzt wird.

[0151] Wenn die Steuerleitung L-Pegel hat, während ein Target bootet, sendet das Target einen Anforderungscode mit einem auf TRUE gesetzten Fehlererfassungsbit. Damit wird es möglich, die Übertragungseinheit zu starten, nachdem alle Mikrocomputer gestartet worden sind, selbst wenn diese nach einer Rücksetzung Anlaufzeitunterschiede aufweisen.

[0152] Wenn die Steuerleitung L-Pegel hat, während die Steuerung bootet, gibt die Steuerung zusätzlich den Takt an die Taktleitung 2 aus, bis die /BSY-Leitung 3 den H-Pegel annimmt, und sendet dann ein REQ-Byte mit /ER-Bit 0, sobald die /BSY-Leitung 3 den H-Pegel angenommen hat. Ist die /BSY-Leitung 3 auf H-Pegel, wenn die Steuerung bootet, sendet sie einen Anforderungscode mit Fehlererfassungsbit TRUE, bevor sie das /BSY-Signal aufprägt. Dadurch können alle Datensendungen, die während der Übertragungseinheit, in der die Steuerung zurückgesetzt wird, erfolgen, zuverlässig ungültig gemacht werden.

[0153] Das einen Datenblock sendende Target bringt die Datenleitung auf L-Pegel, wenn die Sendevorbereitungen für die Datentransfereinheit mit dem Datenblock beendet sind.

[0154] Die Steuerung gibt den Einheitstakt aus, nachdem sie das /BSY-Signal erneut aufgeprägt hat, wenn die Datenleitung zum Zeitpunkt der Einheitstaktausgabe nach Beendigung des Datenblock- oder Nullblocksendens L-Pegel hat und die /BSY-Leitung 3 auf L-Pegel bleibt, selbst wenn die Steuerung das /BSY-Signal negiert hat. Daher kann die Steuerung den Einheitstakt ausgeben, nachdem sie bestätigt hat, dass das Target die Datensendevorbereitungen abgeschlossen hat.

[0155] Es sei darauf hingewiesen, dass bei dem vorstehend beschriebenen Ausführungsbeispiel zwar Mikrocomputer als die an den seriellen Bus angeschlossenen Elemente verwendet werden, dass die Elemente jedoch nicht auf Mikrocomputer beschränkt sind und auch andere LSI-Elemente, beispielsweise Prozessoren für digitale Signale, an den seriellen Bus angeschlossen werden können.

[0156] Beim vorstehenden Ausführungsbeispiel wird als Fehlererfassungscode außerdem ein Paritätsprüfcode verwendet, jedoch ist es auch möglich, einen zyklischen Redundanzprüfcode oder einen anderen Fehlererfassungscode zu verwenden.

[0157] Während beim vorstehenden Ausführungsbeispiel jedem Mikrocomputer ein Zeitschlitz in der Sendereihenfolge zugeteilt ist, können einem einzelnen Mikrocomputer auch mehrere Zeitschlitze in der Sendefolge zugeteilt werden, damit er während einer Transfereinheit mehrere Datenblöcke senden kann.

[0158] Außerdem ist der Bytetaktzyklus beim obigen Ausführungsbeispiel konstant, was nicht so sein muss,

wenn der Bytetakt intermittierend ausgegeben wird. Beispielsweise kann für den Datenkopfeingangsvorgang relativ mehr Zeit erforderlich sein als für andere Datenempfangsvorgänge, z. B. Einstellen des Empfangszählers und Datenkopfbewertungsvorgang. In diesen Fällen kann die Steuerung zeitweise den Unterbrechungszyklus des Taktgebers verlängern, um den Targets für das Senden und Empfangen von Datenköpfen mehr Zeit zur Verfügung zu stellen.

[0159] Außerdem beendet die Steuerung beim obigen Ausführungsbeispiel die Unterbrechungsbearbeitung des Taktgebers, ohne das Ende der Bytetakt Ausgabe abzuwarten, und führt bei der nächsten Taktunterbrechung die Empfangsverarbeitung für mit dem zuvor ausgegebenen Bytetakt gesendete Daten aus. Es ist jedoch auch möglich, während der Taktunterbrechung das Ende der Bytetakt Ausgabe abzuwarten und dann den Datenempfangsvorgang auszuführen. Ferner ist es möglich, den Bytetakt bei Taktunterbrechung auszugeben und die Empfangsverarbeitung während der seriellen Unterbrechung auszuführen.

### Patentansprüche

1. Datentransferverfahren zum seriellen Übertragen von Datenframes zwischen einer Steuereinrichtung (10) und einem Target oder mehreren Targets (11, 12), welche wechselseitig mit einem Signalbus verbunden sind, der eine bidirektionale Datenleitung (1) zum seriellen Übertragen von zumindest den Daten, die einen Datenframe enthalten, eine Taktleitung (2) zum Zuführen eines Sendetakts für die von der Steuereinrichtung (10) über die Datenleitung (1) zu den Targets (11, 12) übertragenen Daten und eine bidirektionale Wired-or-Steuerleitung (3) aufweist,

**dadurch gekennzeichnet,**

dass die Steuereinrichtung (10) damit beginnt, einen Datenframe oder einen Nullframe zu der Datenleitung (1) in Abhängigkeit davon zu übertragen, ob oder ob nicht eine Datenübertragungsanforderung für die Steuereinrichtung (10) vorliegt, nach der Ausgabe eines Pegels LOW an die Steuerleitung (3), wenn die Steuerleitung (3) den Pegel HIGH aufweist, und dass die Steuereinrichtung (10) die Ausgabe des Pegels LOW an die Steuerleitung (3) anhält, nachdem das Übertragen des Datenframes oder des Nullframes beendet worden ist, dass die Übertragungsfolge für die Targets (11, 12) vorbestimmt ist, dass die Targets (11, 12) mit einer Übertragungsanforderung für einen Datenframe beginnen, einen Pegel LOW an die Steuerleitung (3) auszugeben, während die Steuerleitung (3) auf dem Pegel LOW durch die Steuereinrichtung (10) gehalten wird, dass, falls die Steuerleitung (3) auf dem Pegel LOW bleibt, sogar wenn die Steuereinrichtung (10) die Ausgabe des Pegels LOW an die Steuerleitung (3) anhält, nach dem das Übertragen des Datenframes oder des Nullframes beendet hat, die Steuereinrichtung (10) den Sendetakt ausgibt, bis die Steuerleitung (3) den Pegel HIGH annimmt, und das Target (11, 12), welches einen Pegel LOW an die Steuerleitung (3) ausgibt, die Übertragung eines Datenframes in Synchronisation mit dem Sendetakt beginnt, nachdem die Steuereinrichtung (10) und alle Targets (11, 12), die in der Übertragungsfolge früher liegen, das Datenframe- oder Nullframeübertragen beendet haben, und die Ausgabe des Pegels LOW an die Steuerleitung (3) beendet, nachdem die Datenframeübertragung beendet ist.

2. Datentransferverfahren nach Anspruch 1, weiterhin dadurch gekennzeichnet, dass die eine Steuereinrichtung (10) und das eine Target oder die mehreren Targets (11, 12) mit der bidirektionalen Datenleitung (1) für das serielle Übertragen von Daten, die zumindest einen Datenframe bilden, in einer Transfereinheit verbunden sind,

dass die Taktleitung (2) intermittierend als Sendetakte Einheitstakte, die zum Übertragen einer Transfereinheit an Daten zu einer vorbestimmten Ausgabezeit erforderlich sind, von der Steuereinrichtung (10) zu dem einen Target oder den mehreren Targets (11, 12) zuführt,

dass die Steuereinrichtung (10) einen Datenframe oder einen Nullframe über die Datenleitung (1) in Synchronisation mit den Einheitstakten überträgt,

dass die Steuereinrichtung (10) nach Wiederaufnahme der Ausgabe des Pegels LOW an die Steuerleitung (3) den Einheitstakt ausgibt, falls die Steuerleitung (3) auf dem Pegel LOW bleibt, sogar wenn die Steuereinrichtung (10) eine Ausgabe des Pegels LOW an die Steuerleitung (3) zu einer Ausgabezeit des Einheitstaktes unterbricht, nachdem die Übertragung des Datenframes oder des Nullframes beendet worden ist, und

dass jedes Target (11, 12) die Ausgabe des Pegels LOW an die Steuerleitung (3) durch die nächste Ausgabezeit eines Einheitstaktes unterbricht, nachdem die Übertragung des Datenframes beendet ist.

3. Datentransferverfahren nach Anspruch 2,

bei dem das Target, welches einen Datenframe überträgt, die Datenleitung auf den Pegel LOW ändert, wenn die Übertragungsvorbereitungen für eine Transfereinheit an Daten, die einen Datenframe enthält, beendet sind,

bei dem die Steuereinrichtung den Einheitstakt ausgibt, wenn die Datenleitung zu der Einheitstakt-Ausgabezeit auf den Pegel LOW gesetzt ist, und

bei dem das Target eine Transfereinheit an Daten, die mit dem Einheitstakt synchronisiert sind, überträgt und die Datenleitung auf den Pegel HIGH ändert, wenn sie die Datenübertragung beendet.

4. Datentransferverfahren nach Anspruch 1 oder 3,  
bei dem der Nullframe eine feste Nullheaderlänge besitzt,  
bei dem der Datenframe eine feste Datenheaderlänge und variable Nutzerdatenlängen enthält,  
bei dem der Datenheader Daten enthält, die die Art und die Länge der Nutzerdaten kennzeichnen,  
bei dem das Target (11, 12), welches einen Pegel LOW an die Steuerleitung (3) ausgibt, das Datenframeende durch Zählen der Datenlänge der Nutzerdaten erfasst und das Ende des Datenframes oder des Nullframes, welcher durch die Steuereinrichtung (10) und sämtlichen Targets (11, 12), die in der Übertragungsfolge früher liegen, durch Zählen der Zahl der Datenframes und Nullframes erfasst.

5. Datentransferverfahren nach Anspruch 4,  
bei dem, während die Steuerleitung (3) den Pegel LOW einnimmt, die Steuereinrichtung (10) einen Fehlererfassungscode für alle Daten, die über die Datenleitung (1) übertragen worden sind, unter Verwendung der durch die Steuereinrichtung (10) über die Datenleitung (1) übertragenen Daten und der durch die Steuereinrichtung (10) von der Datenleitung (1) empfangenen Daten berechnet,  
und den Fehlererfassungscode ausgibt, nachdem die Steuerleitung (3) den Pegel HIGH einnimmt,  
und alle Targets (11, 12) Übertragungsfehler unter Verwendung der durch das entsprechende Target über die Datenleitung (1) übertragenen Daten, den von der Datenleitung (1) empfangenen Daten und dem empfangenen Fehlererfassungscode erfasst,  
und zu einer vorbestimmten Zeit einen Anforderungscode überträgt, in dem zumindest ein Fehlererfassungsbit auf TRUE gesetzt ist, wenn ein Fehler erfasst wird.

6. Datentransferverfahren nach Anspruch 5,  
bei dem der Datenheader weiterhin einen Fehlererfassungscode zum Erfassen von Übertragungsfehlern in dem Datenheader enthält,  
bei dem die Steuereinrichtung (10) einen Anforderungscode mit einem TRUE-Fehlererfassungsbit überträgt, wenn ein Übertragungsfehler in dem Datenheader erfasst wird, und  
bei dem die Targets (11, 12) die Ausgabe unterbrechen, wenn ein Pegel LOW an die Steuerleitung (3) ausgegeben wird, falls ein Übertragungsfehler in dem Datenheader erfasst wird, und einen Anforderungscode mit einem TRUE-Fehlererfassungsbit übertragen.

7. Datentransferverfahren nach Anspruch 5,  
bei dem Steuereinrichtung (10) und die Targets (11, 12) einen Anforderungscode mit zumindest einem TRUE-Datenframe-Anforderungsbit überträgt, wenn ein Datenframe, der empfangen werden sollte, nicht empfangen wird, und  
bei dem, wenn das Datenframe-Anforderungsbit des empfangenen Anforderungscodes dem Zustand TRUE aufweist, die Steuereinrichtung (10) und jedes Target (11, 12) alle Datenframes sequenziell senden, die jede Einrichtung senden sollte, bis ein Anforderungscode mit einem FALSE-Datenframe-Anforderungsbit empfangen wird.

8. Datentransferverfahren nach Anspruch 5, bei dem die Steuereinrichtung (10) ein Anforderungscode mit einem TRUE-Fehlererfassungsbit sendet, wenn die Steuerleitung (3) einen Pegel HIGH einnimmt, bevor das übertragende Target sämtliche Daten, die durch die Nutzerdaten-Längenspezifikation des Datenheaders gekennzeichnet sind, überträgt.

9. Datentransferverfahren nach Anspruch 5, bei dem die Targets (11, 12) einen Anforderungscode mit einem TRUE-Fehlererfassungsbit übertragen, wenn die Steuerleitung (3) einen Pegel LOW einnimmt, falls das Target bootet.

10. Datentransferverfahren nach Anspruch 5,  
bei dem Steuereinrichtung (10) den Takt an die Taktleitung (2) ausgibt, bis die Steuerleitung (3) den Pegel HIGH einnimmt, und einen Anforderungscode mit einem TRUE-Fehlererfassungsbit überträgt, falls die Steuerleitung (3) den Pegel LOW einnimmt, wenn die Steuereinrichtung bootet, und,  
falls die Steuerleitung (3) den Pegel HIGH einnimmt, wenn die Steuereinrichtung (10) bootet, die Steuereinrichtung (10) ein Anforderungscode mit einem TRUE-Fehlererfassungsbit zumindest zweimal ausgibt, bevor ein Pegel LOW an die Steuerleitung (3) ausgegeben wird.

11. Datentransfersystem, enthaltend eine Vielzahl an Mikrocomputern, von denen einer als eine Steuerein-



richtung (10) bezeichnet wird, und die übrigen als Targets (11, 12) bezeichnet werden, und einen seriellen Datenbus, der eine bidirektionale Datenleitung (1) enthält, über die ein Übertragen von Daten und ein Empfangen von Daten ausgeführt wird, eine Taktleitung (2), welche Taktsignale, die durch die Steuereinrichtung erzeugt werden, an jedes Target (11, 12) überträgt, und eine bidirektionale Wired-or-Steuerleitung (3), welche sowohl für die Steuereinrichtung (10) als auch für die Targets (11, 12) zugreifbar und ablehnbar ist, dadurch gekennzeichnet,

dass die Steuereinrichtung (10) einen Datenframe oder einen Nullframe an die Datenleitung (1) in Abhängigkeit davon überträgt, ob oder ob nicht eine Datenübertragungsanforderung für die Steuereinrichtung besteht, nachdem die Steuereinrichtung (10) auf die Steuerleitung (3) zugreift, wenn die Steuerleitung (3) den Pegel HIGH einnimmt, und, falls die Steuerleitung (3) auf dem Pegel LOW sogar bleibt, wenn die Steuereinrichtung (10) die Ausgabe des Pegels LOW an die Steuerleitung (3) unterbricht, nachdem das Übertragen des Datenframes oder des Nullframes beendet worden ist, die Steuereinrichtung (10) das Taktsignal ausgibt, bis die Steuerleitung (3) den Pegel HIGH einnimmt, und dass jedes Target (11, 12), das eine eigene Sendefolge an Daten, die vorher gesammelt worden sind, besitzt, eine Ausgabe für das Zugreifen auf die Steuerleitung (3) ausgibt, während die Steuerleitung (3) in einem Zugangszustand durch die Steuereinrichtung (10) gehalten wird, falls eine Sendeanforderung eines Datenframes besteht, und, wenn die Steuereinrichtung (10) und die vorausgegangenen Targets (11, 12) das Datenframe- oder Nullframeübertragen beendet haben, eine Datenframeübertragung in Synchronisation mit den Taktsignalen von der Steuereinrichtung (10) beginnen und, falls sie nicht irgendeine Übertragungsanforderung eines Datenframes besitzt, einen Nullframe an die Datenleitung (1) ausgibt, ohne dass eine Ausgabe zum Zugreifen der Steuerleitung (3) ausgegeben wird, und wenn das Frameübertragen beendet worden ist, die Ausgabe für den Zugriff auf die Steuerleitung (3) unterbricht.

Es folgen 19 Blatt Zeichnungen



Fig. 1

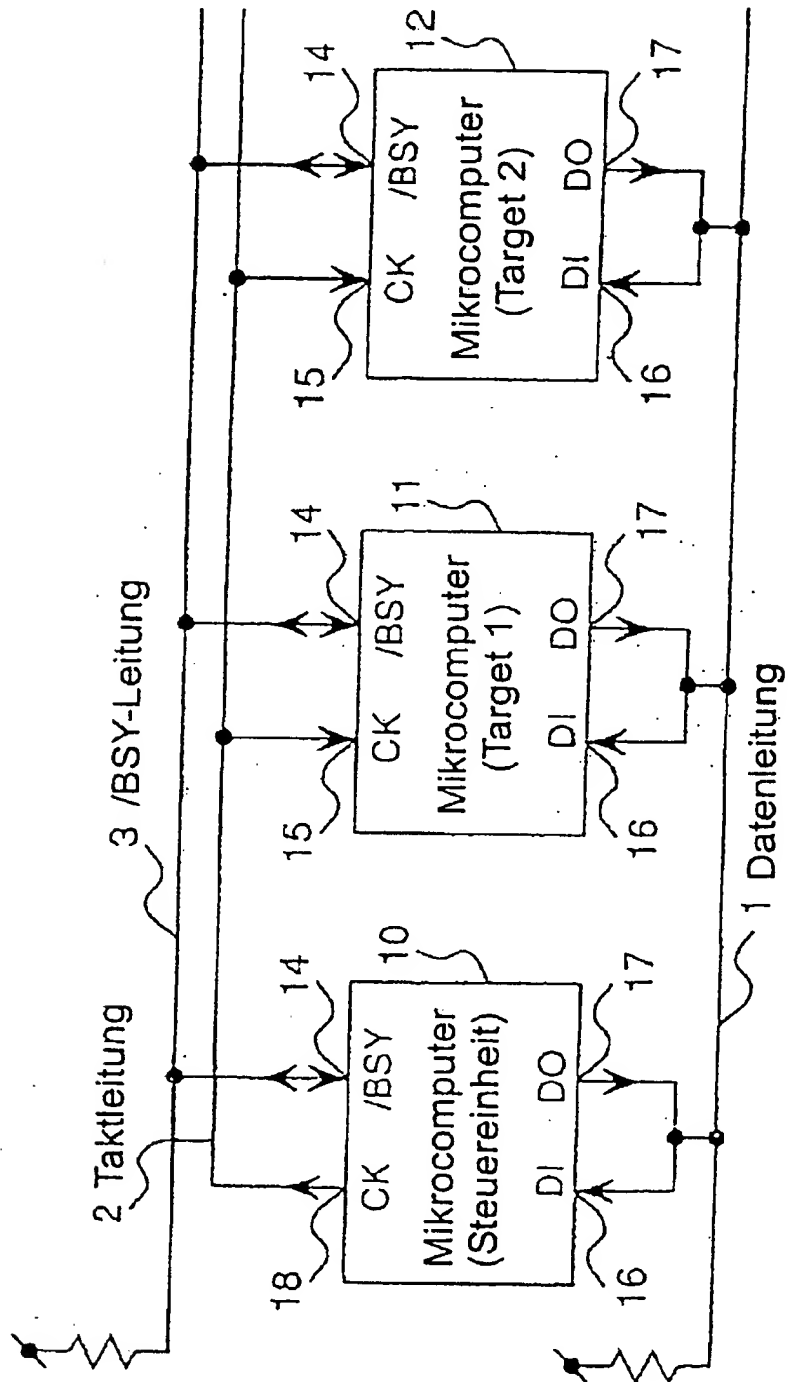


Fig.2

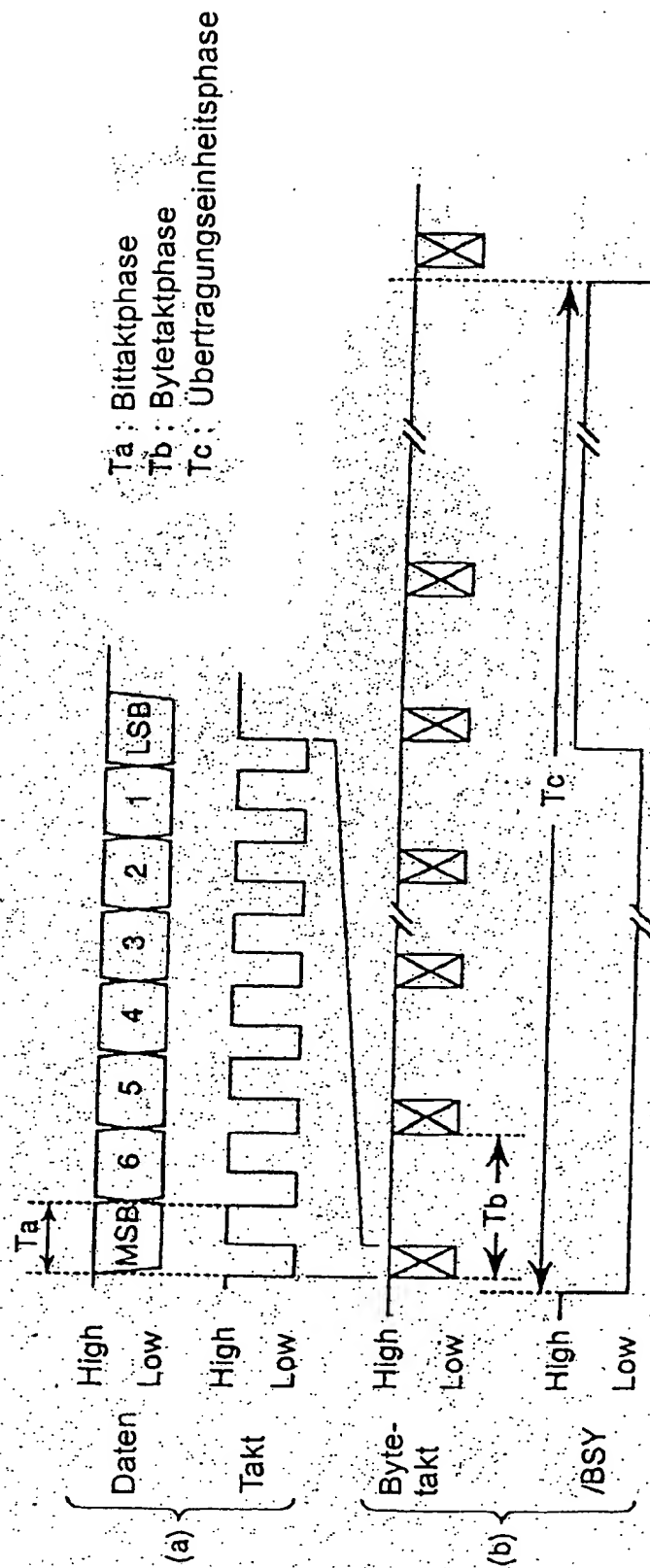


Fig.3

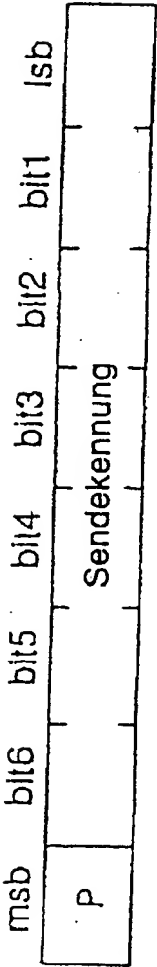


Fig.4

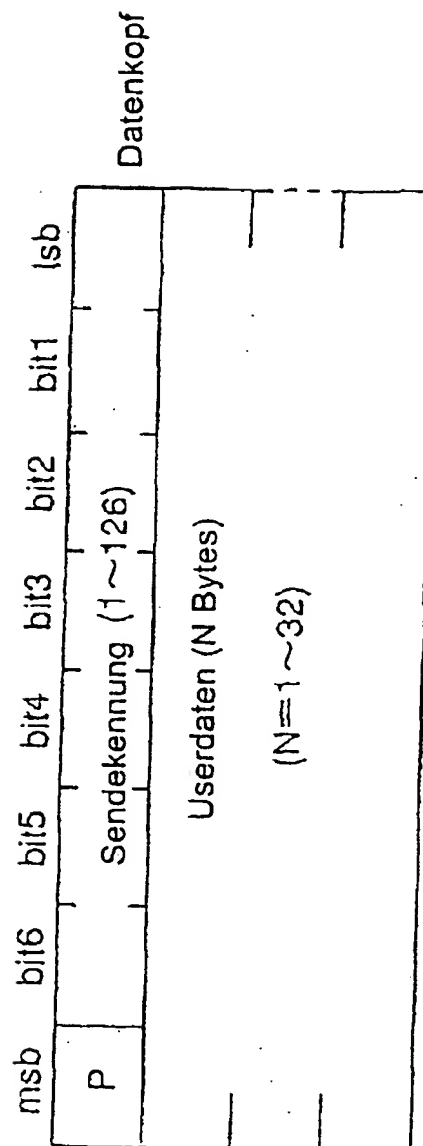


Fig.5

	msb	bit6	bit5	bit4	bit3	bit2	bit1	lsb	
Gruppe A	P	0	X			N-1			N=1~32
Gruppe B	P	1	0	X		N-1			N=1~16
Gruppe C	P	1	1	X	X		N-1		N=1~8

Fig. 6

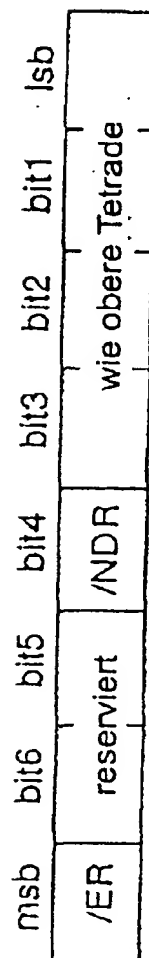
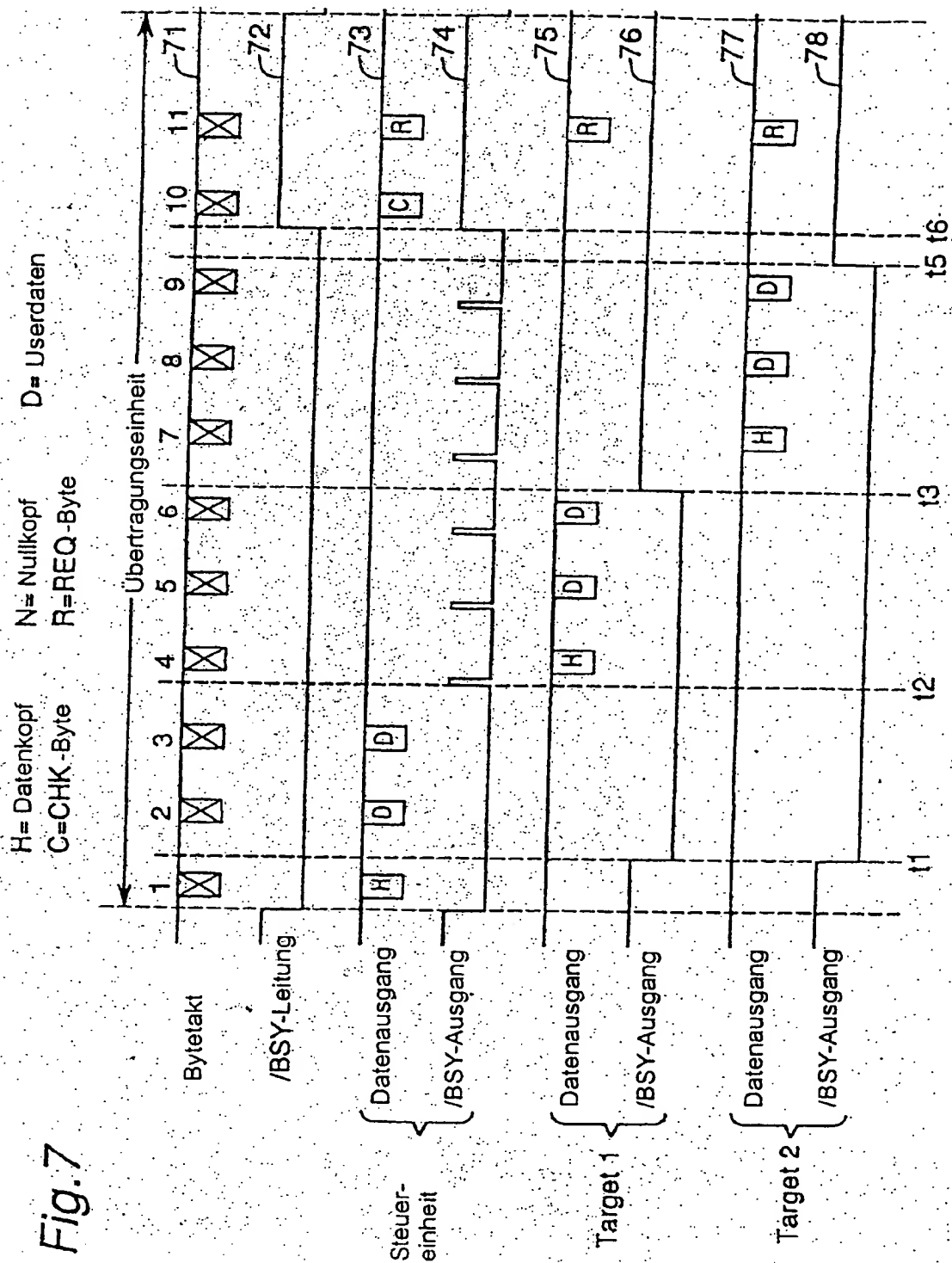


Fig. 7



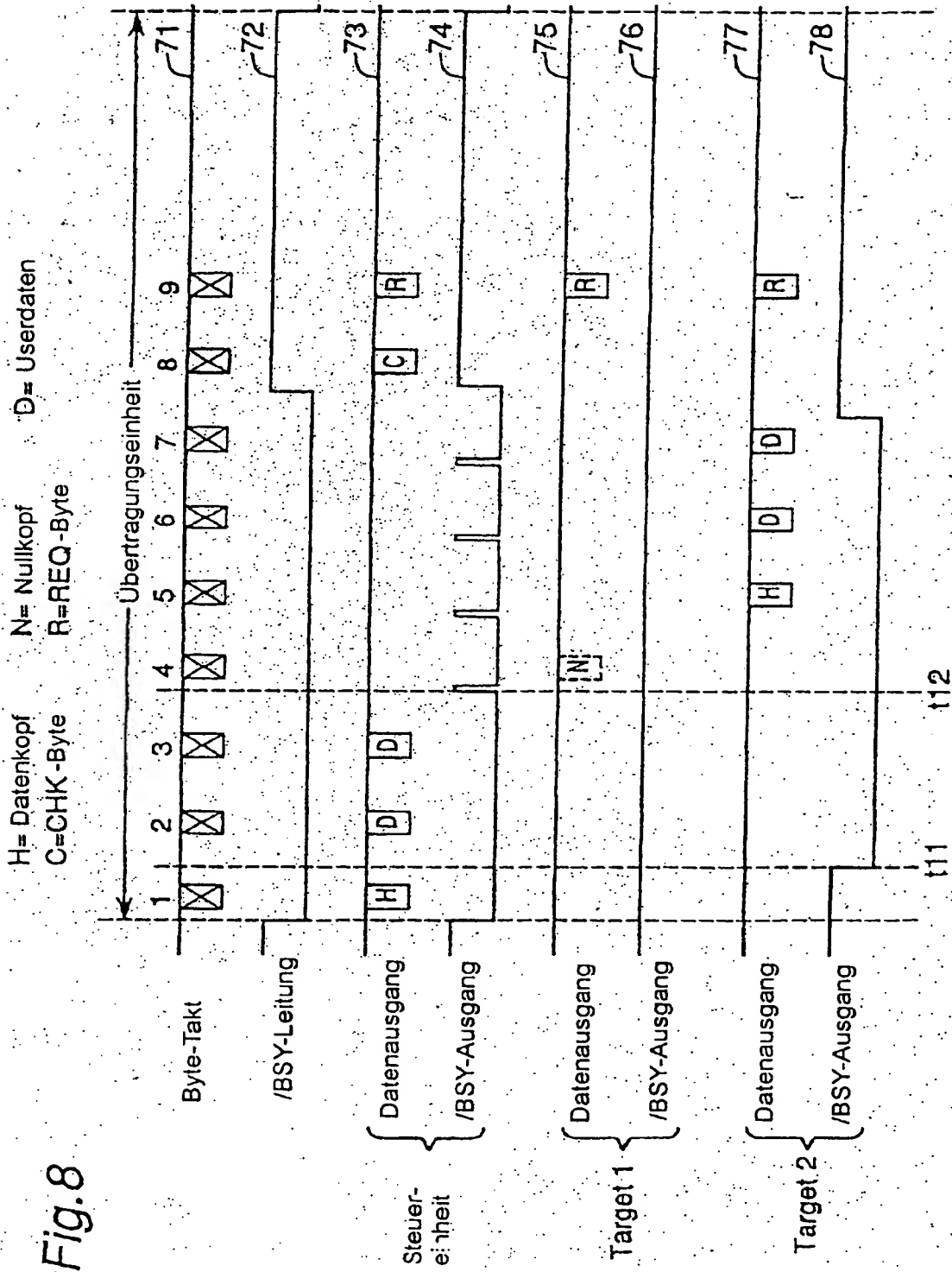
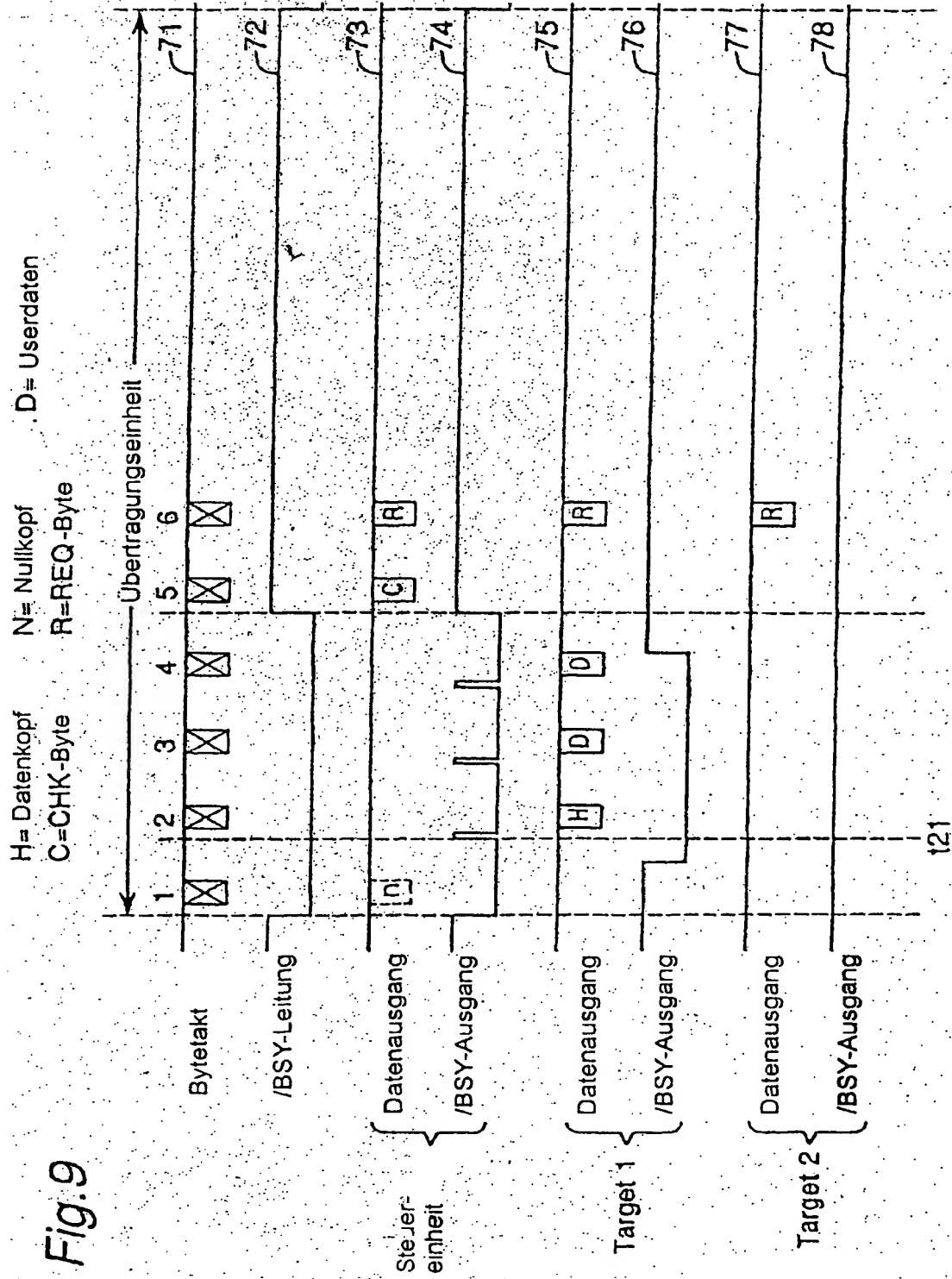




Fig.9



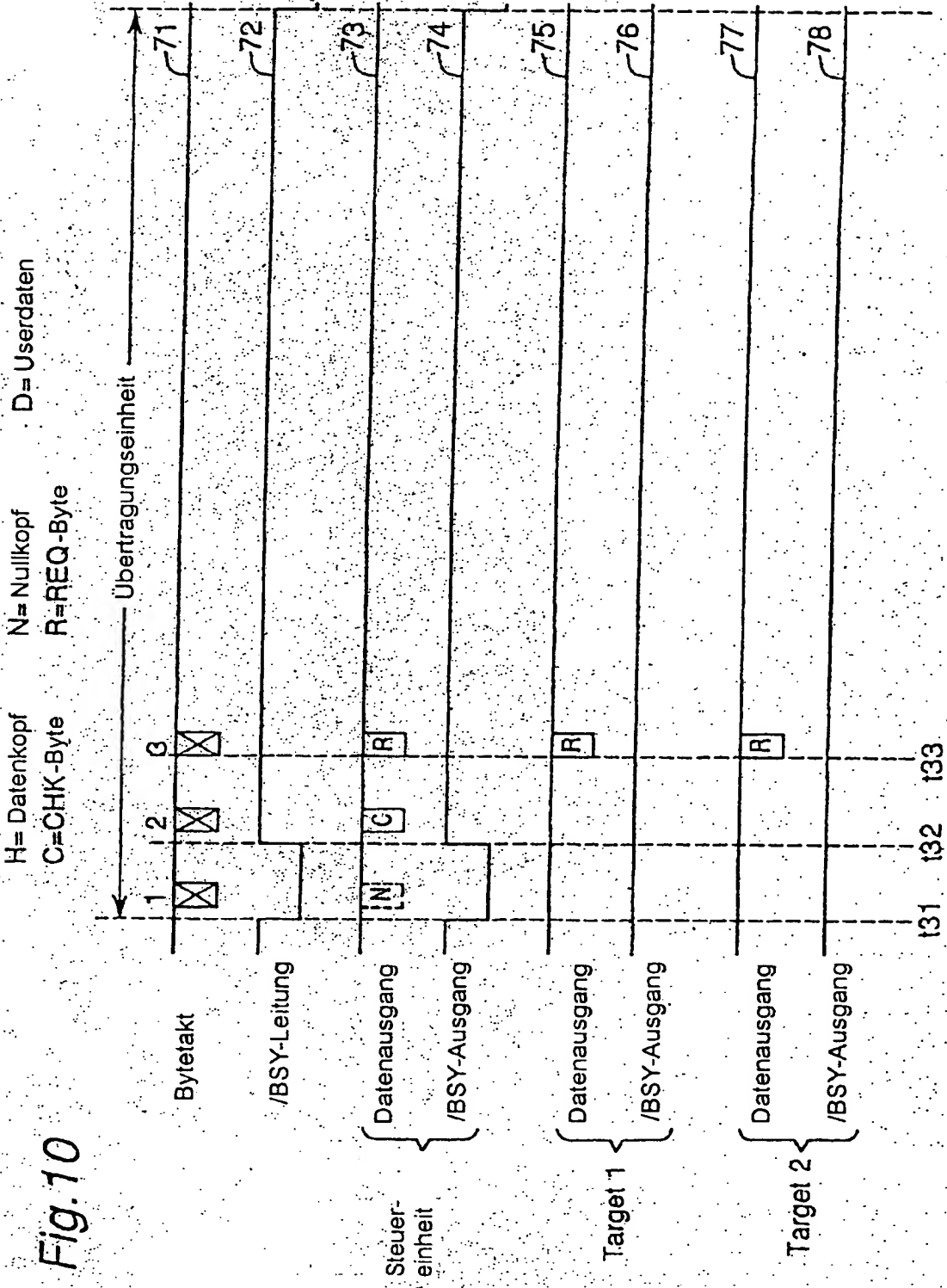


Fig. 11A

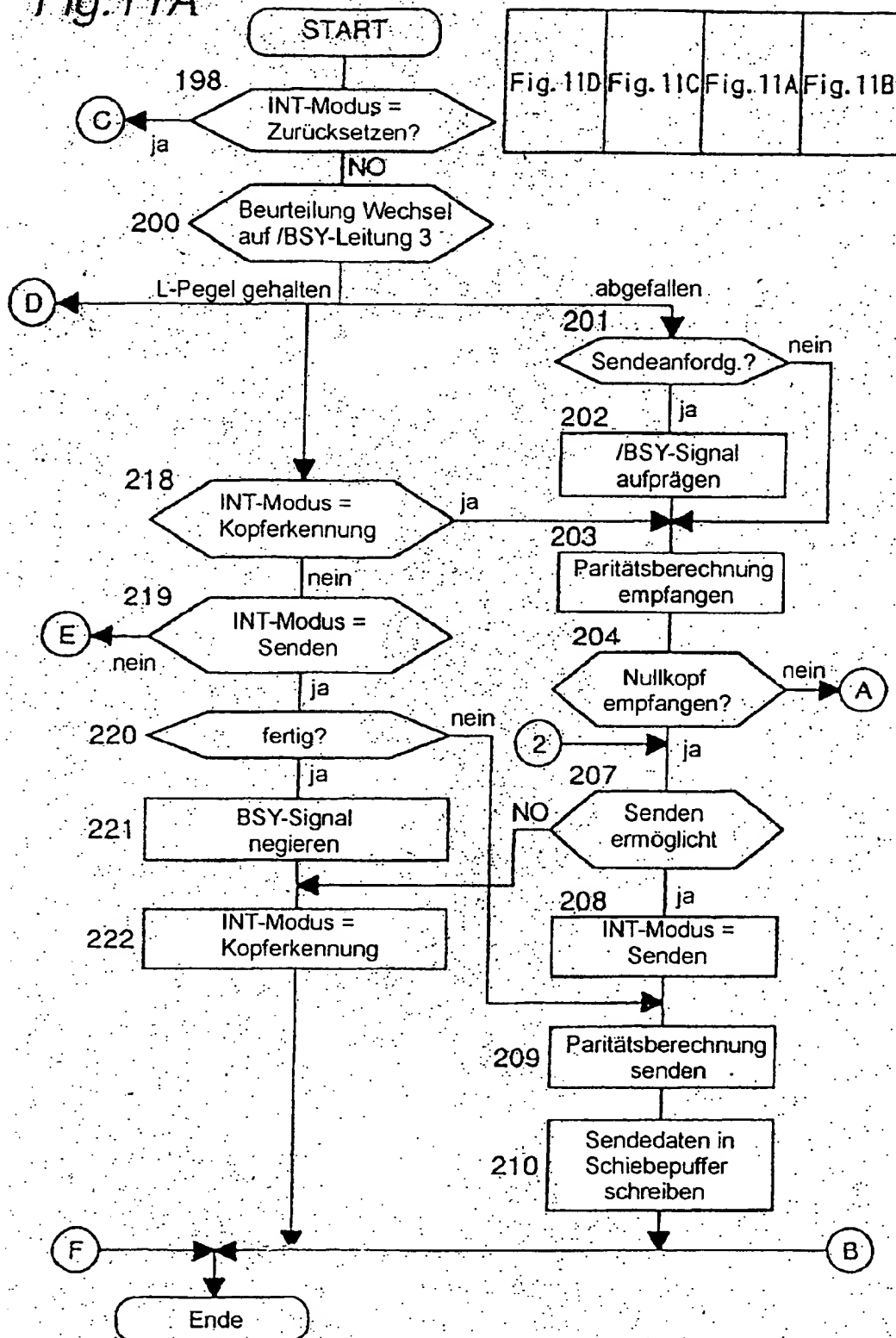


Fig. 11B

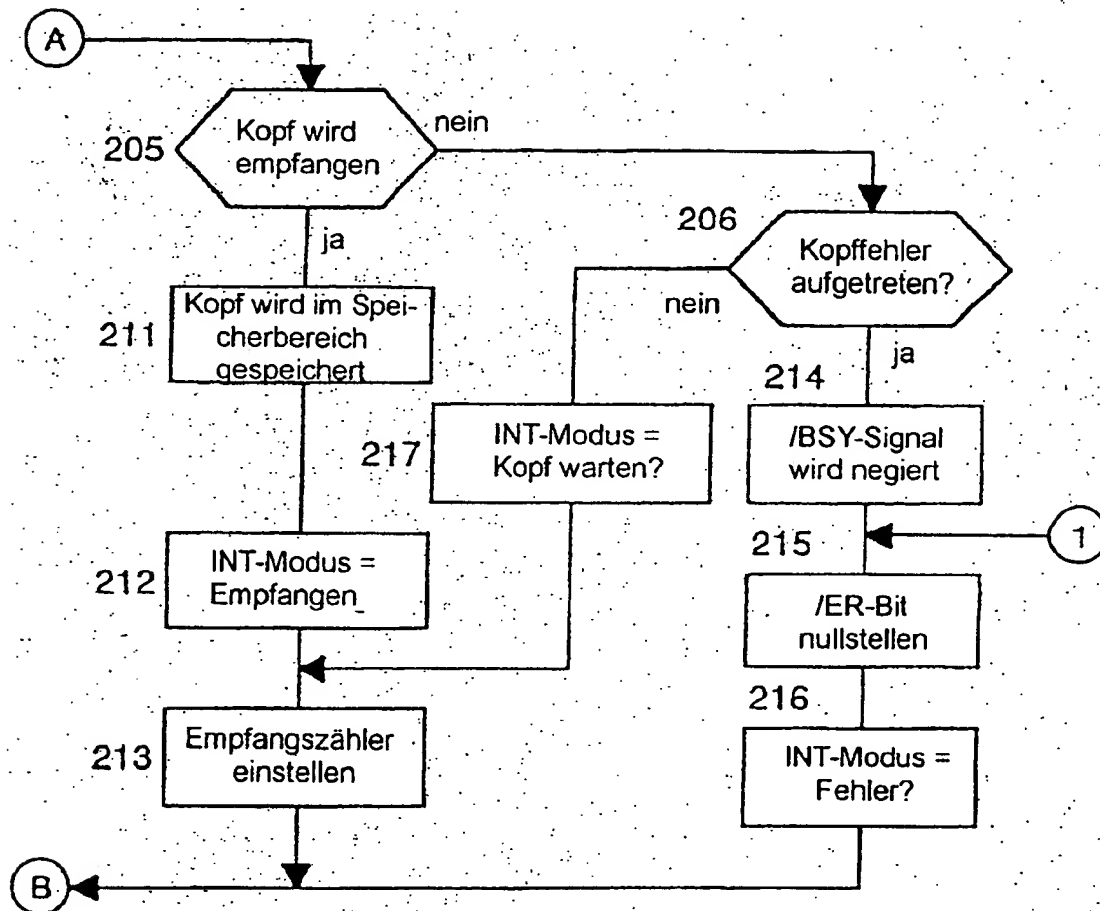


Fig. 11C

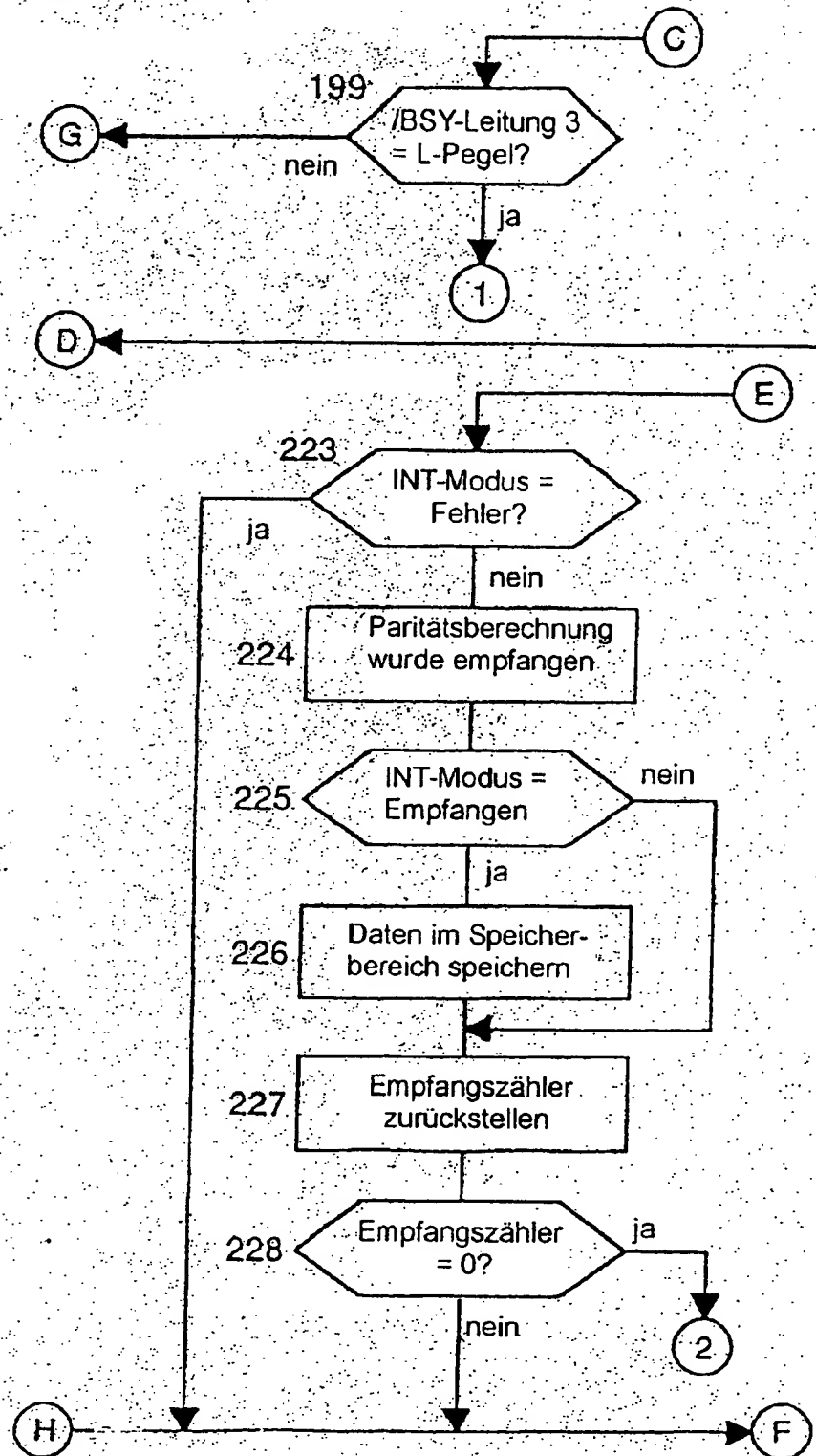
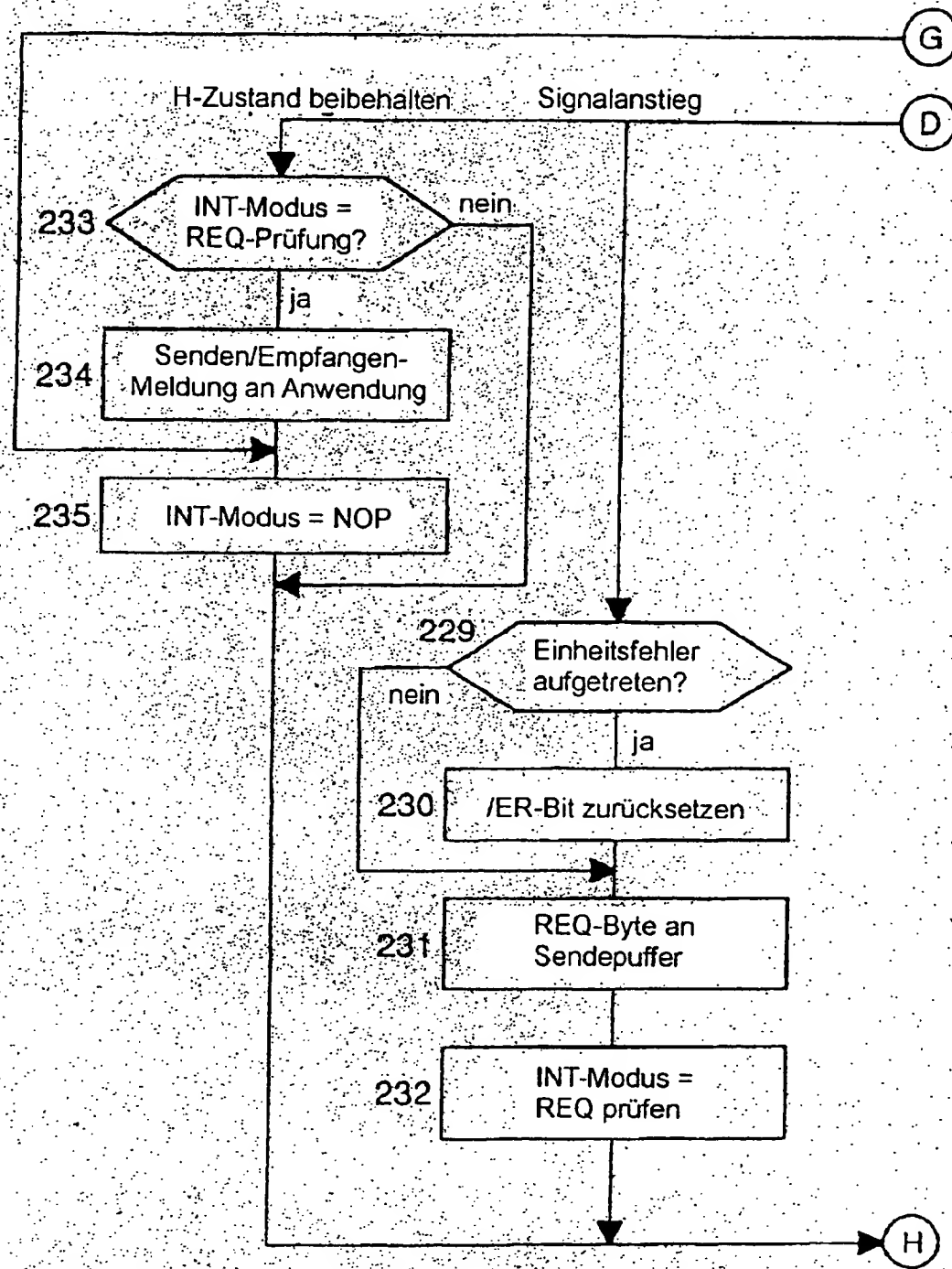


Fig. 11D



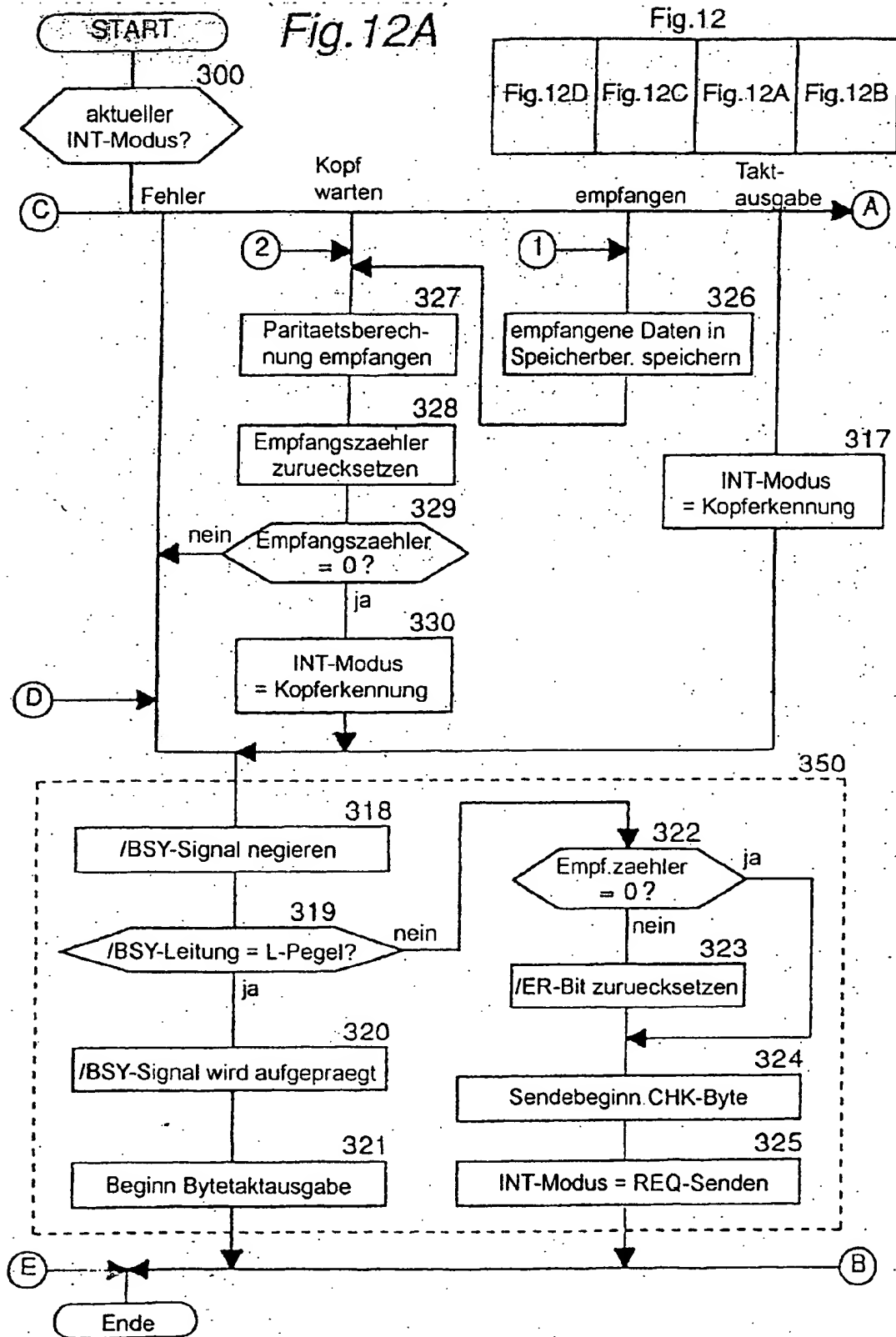


Fig.12B

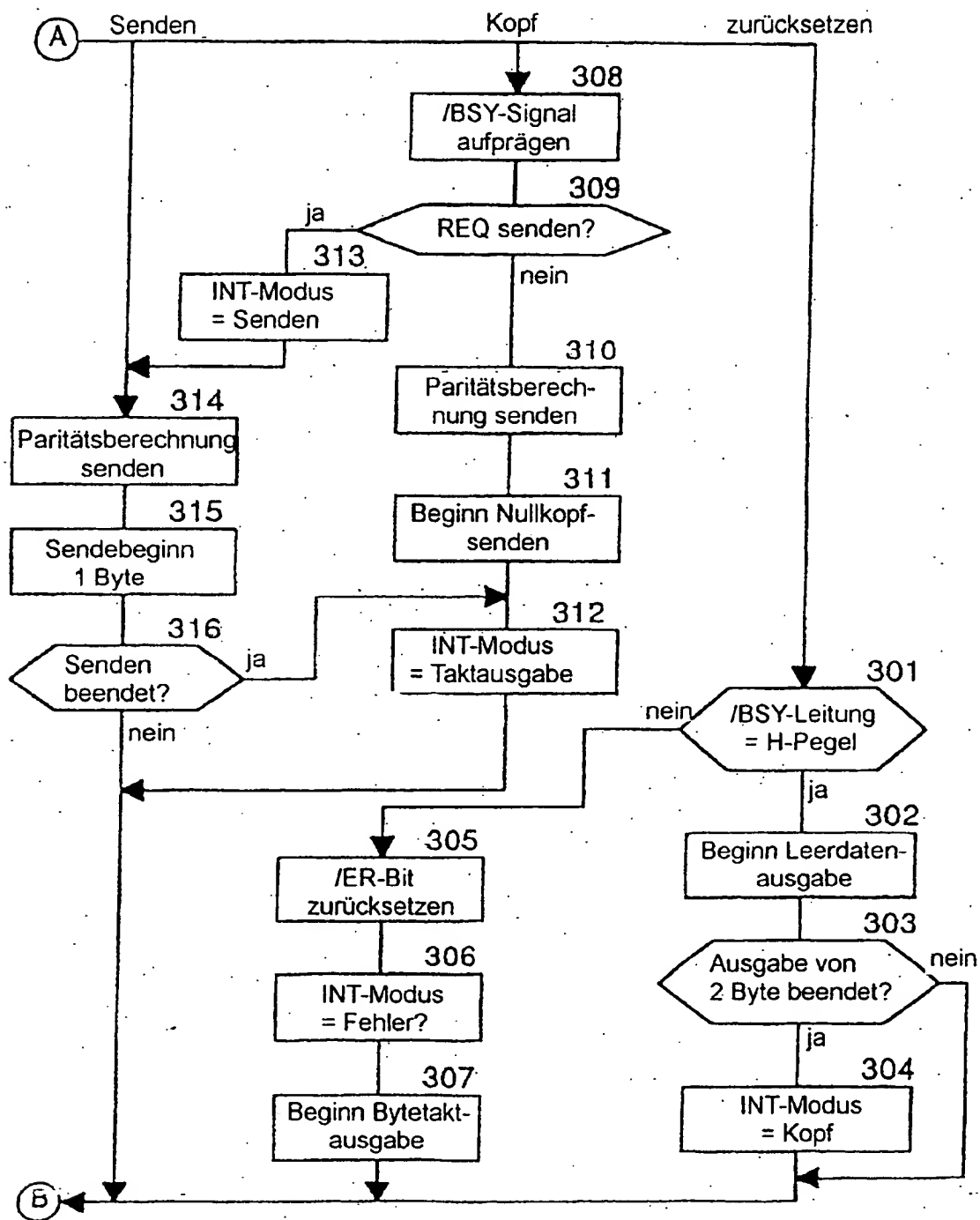




Fig.12C

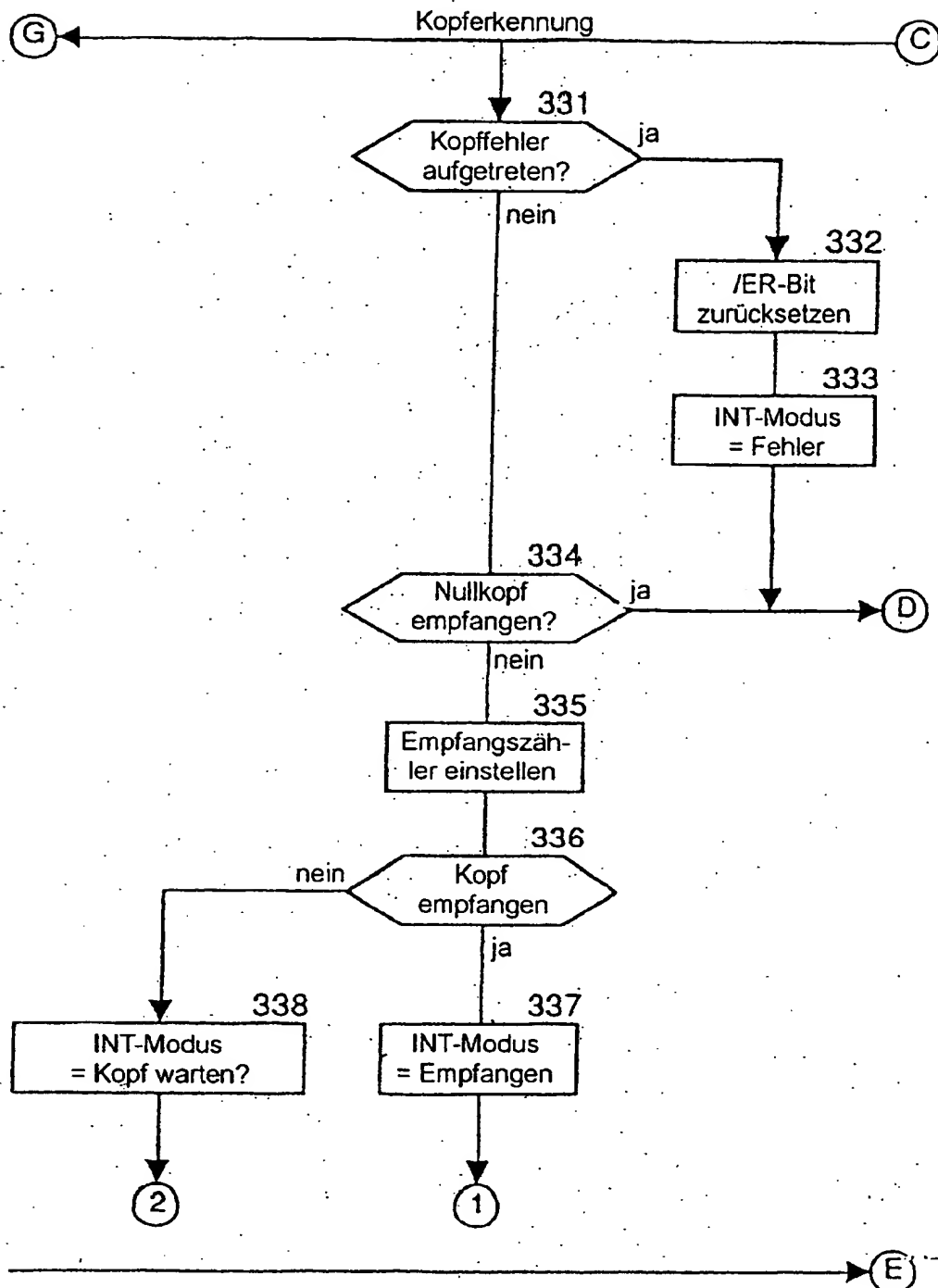


Fig. 12D

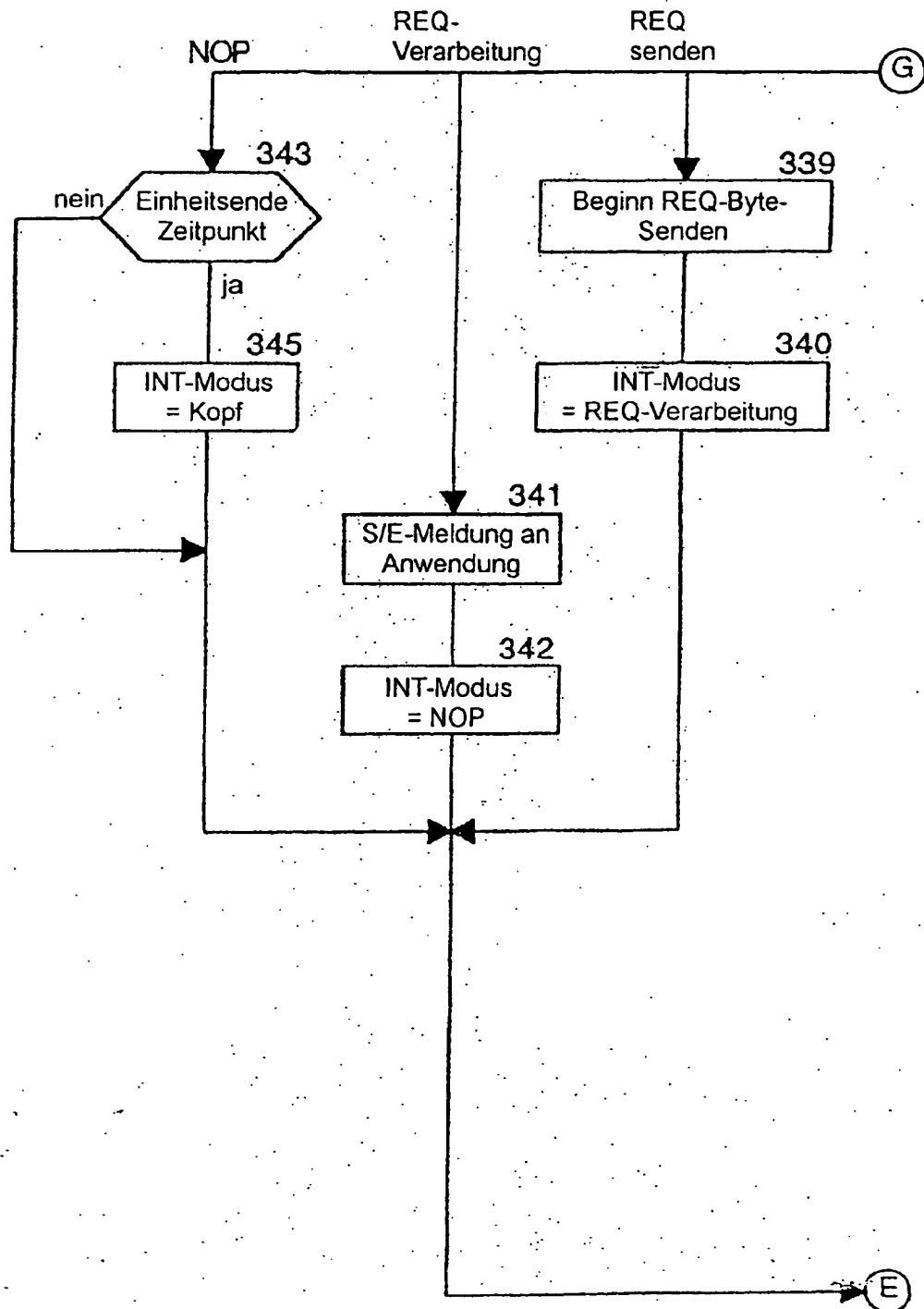
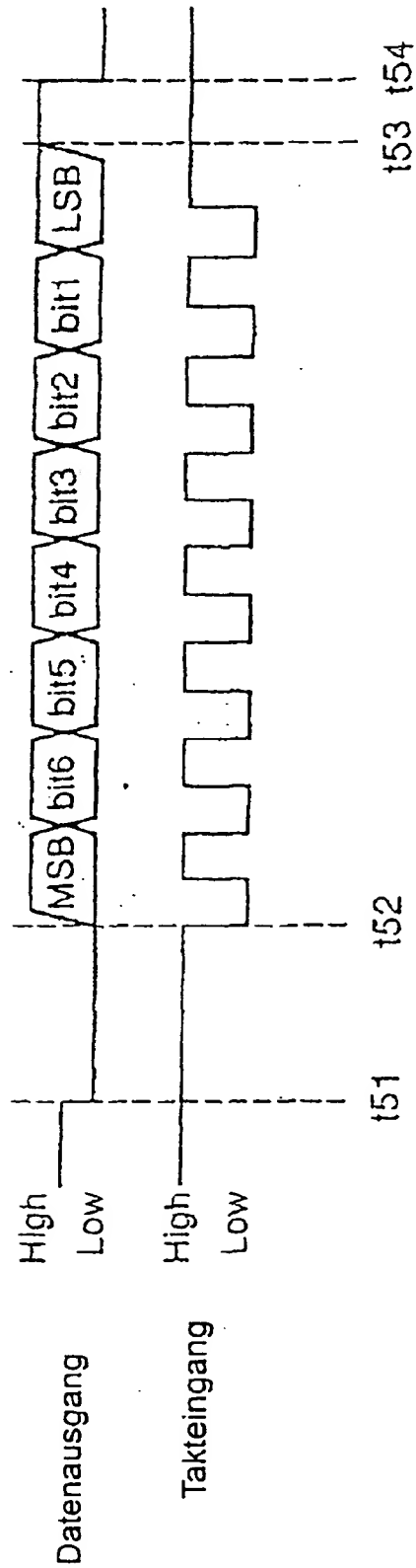


Fig.13



4/9/1

DIALOG(R) File 351:Derwent WPI

(c) 2007 The Thomson Corporation. All rts. reserv.

0007017167 - Drawing available

WPI ACC NO: 1995-031795/199505

XRPX Acc No: N1995-025298

**Appts for serial data transfer between microcomputers - has controller checking busy line confirmation and sending or receiving data while it is**

**acknowledged with salves operating in predefined order**

Patent Assignee: MATSUSHITA ELEC IND CO LTD (MATU); MATSUSHITA ELECTRIC

IND CO LTD (MATU)

Inventor: IITSUKA H; NISHIMURA T

**Patent Family** (4 patents, 3 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
EP 631239	A2	19941228	EP 1994107528	A	19940516	199505 B
US 5600803	A	19970204	US 1994242441	A	19940513	199711 E
EP 631239	B1	20030528	EP 1994107528	A	19940516	200336 E
DE 69432726	E	20030703	DE 69432726	A	19940516	200351 E
			EP 1994107528	A	19940516	

Priority Applications (no., kind, date): EP 1994107528 A 19940516; JP 1993112860 A 19930514

**Patent Details**

Number	Kind	Lan	Pg	Dwg	Filing Notes
EP 631239	A2	EN	41	13	
Regional Designated States, Original: DE FR GB					
US 5600803	A	EN	32	13	
EP 631239	B1	EN			
Regional Designated States, Original: DE FR GB					
DE 69432726	E	DE			Application EP 1994107528
					Based on OPI patent EP 631239

**Alerting Abstract** EP A2

The system includes a serial bus for transferring data. The microcomputers (10,11,12) are linked by a serial transfer line (1), a clock

line (2) and a busy line (3). One microcomputer is the controller and the

others are slave unit responding in predefined order.

The controller asserts the busy line if it is not already busy and transfers data. Slaves can jointly assert the busy line and the controller

check this by temporarily change line status to see if it is still asserted. If so a slave is allowed to transfer data to the controller.

USE/ADVANTAGE - For transferring data frames between single-chip computers. Improves efficiency of data transfer by avoiding use of interrupts or busy collisions.

**Title Terms** /Index Terms/Additional Words: APPARATUS; SERIAL; DATA; TRANSFER

; MICROCOMPUTER; CONTROL; CHECK; BUSY; LINE; CONFIRM; SEND; RECEIVE; ACKNOWLEDGE; SALVE; OPERATE; PREDEFINED; ORDER

#### Class Codes

International Classification (Main): G06F-013/00, G06F-013/36  
US Classification, Issued: 395287000, 395856000, 395855000, 395850000,  
395200050, 364222200, 364283300, 364238300, 364DIG001

File Segment: EPI;

DWPI Class: T01

Manual Codes (EPI/S-X): T01-H05B3

#### Original Publication Data by Authority

##### Germany

Publication No. DE 69432726 E (Update 200351 E)

Publication Date: 20030703

Assignee: MATSUSHITA ELECTRIC IND CO LTD; JP (MATU)

Language: DE

Application: DE 69432726 A 19940516 (Local application)

EP 1994107528 A 19940516 (Application)

Priority: JP 1993112860 A 19930514

Related Publication: EP 631239 A (Based on OPI patent )

##### EPO

Publication No. EP 631239 A2 (Update 199505 B)

Publication Date: 19941228

**\*\*Verfahren zur seriellen Datenuebertragung**

Serial data transfer method

Methode de transfert de donnees serie\*\*

Assignee: MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD., 1006, Oaza Kadoma,  
Kadoma-shi, Osaka-fu, 571, JP (MATU)

Inventor: Iitsuka, Hiroyuki, 5-9-5-305, Kouzu, Katano-shi, Osaka, JP

Nishimura, Takuya, 6-1-105, Myokenzaka, Katano-shi, Osaka, JP

Agent: Eisenfuehr, Speiser Partner, Martinistrasse 24, D-28195 Bremen,  
DE

Language: EN (41 pages, 13 drawings)

Application: EP 1994107528 A 19940516 (Local application)

Priority: JP 1993112860 A 19930514

Designated States: (Regional Original) DE FR GB

Original IPC: G06F-13/36(A)

Current IPC: G06F-13/36(A)

Original Abstract: A data transfer system including a plurality of  
microcomputers connected to a serial data bus in parallel to each  
other, one of which is designated as a controller and remainder as  
targets each having its own sending order of data allocated

beforehand,

in which the controller asserts a control line of the serial data  
bus

to start a data frame or a null frame sending to a data line of the  
serial data bus and each target outputs an output for asserting the  
control line during the control line is kept in an asserted state

by

the controller if it has a data frame sending request, and outputs

a

data frame or a null frame when its sending order has come.

Claim: 1. A data transfer method for serially sending data frames  
between

one controller and one or more targets for which the sending order  
 is  
 predetermined and which are mutually connected to a signal bus  
 comprising a bidirectional data line for serially sending in  
 single  
 transfer units at least the data comprising a data frame, a clock  
 line for intermittently supplying the unit clock required for  
 sending  
 one transfer unit of data at a predetermined output timing from the  
 controller to the targets, and a bidirectional control line,  
 and  
 characterized by the controller synchronizing data frame or null  
 frame  
 sending to the unit clock after beginning LOW level output to the  
 control line when the control line is HIGH, targets with a data  
 frame  
 send request beginning LOW level output to the control line when  
 the  
 control line is held LOW by the controller, the controller  
 outputting  
 the unit clock after resuming LOW level output to the control line  
 if  
 the control line remains LOW even though the controller stops LOW  
 output to the control line at the unit clock output timing after  
 data  
 frame or null frame sending ends, and the target outputting a  
 LOW  
 level to the control line beginning data frame sending synchronized  
 to  
 the unit clock after the controller and all targets earlier in the  
 sending order complete data frame or null frame sending,  
 \* and stopping LOW output to the control line by the next unit  
 clock  
 output timing after it completes data frame sending.

Publication No. EP 631239 B1 (Update 200336 E)

Publication Date: 20030528

**\*\*Verfahren und System zur seriellen Datenübertragung**

Serial data transfer method and system

Methode et systeme de transfert de donnees serie\*\*

Assignee: MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD., 1006, Oaza-Kadoma,  
 Kadoma-shi, Osaka 571-8501, JP (MATU)

Inventor: Iitsuka, Hiroyuki, 5-9-5-305, Kouzu, Katano-shi, Osaka, JP  
 Nishimura, Takuya, 6-1-105, Myokenzaka, Katano-shi, Osaka, JP

Agent: Eisenfuhr, Speiser Partner, Martinistrasse 24, 28195 Bremen, DE  
 Language: EN

Application: EP 1994107528 A 19940516 (Local application)

Designated States: (Regional Original) DE FR GB

Original IPC: G06F-13/36(A)

Current IPC: G06F-13/36(A)

Claim: Datentransferverfahren zum seriellen Übertragen von  
 Datenframes

zwischen einer Steuereinrichtung (10) und einem Target oder  
 mehreren

Targets (11, 12), welche wechselseitig mit einem Signalbus  
 verbunden

sind, der eine bidirektionale Datenleitung (1) zum seriellen Übertragen von zumindest den Daten, die einen Datenframe enthalten, eine Taktleitung (2) zum Zuführen eines Sendetakts für die von der Steuereinrichtung (10) über die Datenleitung (1) zu den Targets (11, 12) übertragenen Daten und eine bidirektionale Wired-or-Steuerleitung (3) aufweist,

1.

**\*\*dadurch gekennzeichnet, dass\*\*** die Steuereinrichtung (10) damit beginnt, einen Datenframe oder einen Nullframe zu der Datenleitung (1) in Abhängigkeit davon zu übertragen, ob oder ob nicht eine Datenübertragungsanforderung für die Steuereinrichtung (10) vorliegt, nach der Ausgabe eines Pegels LOW an die Steuerleitung (3), wenn die Steuerleitung (3) den Pegel HIGH aufweist, und dass die Steuereinrichtung (10) die Ausgabe des Pegels LOW an die Steuerleitung (3) anhält, nachdem das Übertragen des Datenframes oder des Nullframes beendet worden ist,

2.

**\*\*dass\*\*** die Übertragungsfolge für die Targets (11, 12) vorbestimmt ist, **\*\*dass\*\*** die Targets (11, 12) mit einer Übertragungsanforderung für einen Datenframe beginnen, einen Pegel LOW an die Steuerleitung (3) auszugeben, während die Steuerleitung (3) auf dem Pegel LOW durch die Steuereinrichtung (10) gehalten wird, **\*\*dass\*\***, falls die Steuerleitung (3) auf dem Pegel LOW bleibt, sogar wenn die Steuereinrichtung (10) die Ausgabe des Pegels LOW an die Steuerleitung (3) anhält, nach dem das Übertragen des Datenframes oder des Nullframes geendet hat, die Steuereinrichtung (10) den Sendetakt ausgibt, bis die Steuerleitung (3) den Pegel HIGH annimmt, und das Target (11, 12), welches einen Pegel LOW an die Steuerleitung (3) ausgibt, die Übertragung eines Datenframes in Synchronisation mit dem Sendetakt beginnt, nachdem die Steuereinrichtung (10) und alle Targets (11, 12), die in der Übertragungsfolge früher liegen, das Datenframe- oder Nullframeübertragen beendet haben, und die Ausgabe des Pegels LOW an die Steuerleitung (3) beendet, nachdem die Datenframeübertragung beendet ist.

A data transfer method for serially sending data frames between one controller (10) and one or more targets (11, 12) which are mutually connected to a signal bus comprising a bidirectional data line (1) for

serially sending at least the data comprising a data frame, a clock line (2) for supplying a send clock for the data sent over the data line (1) from the controller (10) to the targets (11, 12), and a bidirectional wired-or control line (3),

1.  
\*\*characterized in that\*\* the controller (10) starts data frame or null frame sending to said data line (1), according to whether or not a data sending request exists for the controller (10), after outputting a LOW level to the control line (3) when the control line (3) is HIGH, and stopping LOW output to the control line (3) after ending data frame or null frame sending, the sending order is predetermined for the targets (11, 12), targets (11, 12) with a data frame send request begin LOW level output to the control line (3) while the control line (3) is held LOW by the controller (10), if the control line (3) remains LOW even though the controller (10) stops LOW output to the control line (3) after ending data frame or null frame sending, the controller (10) outputs the send clock until the control line (3) becomes HIGH, and the target (11, 12) outputting a LOW level to the control line (3) begins transmission of a data frame in synchronization with the send clock, after the controller (10) and all targets (11, 12) earlier in the sending order complete data frame or null frame sending, and stops LOW output to the control line (3) after it completes data frame transmission.

Une methode de transfert de donnees pour envoyer en serie des trames de donnees entre un dispositif de commande (10) et une ou plusieurs cibles (11, 12) qui sont mutuellement connectes a un bus de signaux comprenant une ligne de donnees bidirectionnelle (1) permettant d'envoyer en serie au moins les donnees constituant une trame de donnees, une ligne d'horloge (2) permettant de fournir une impulsion d'horloge d'envoi pour les donnees envoyees sur la ligne de donnees (1) depuis le dispositif de commande (10) jusqu'aux cibles (11, 12), et une ligne de commande bidirectionnelle cablee en OU (3),

1.  
\*\*caracterisee en ce que\*\* le dispositif de commande (10) commence l'envoi de trame de donnees ou de trame vide a ladite ligne de donnees (1), selon qu'une demande d'envoi de donnees existe ou non pour le dispositif de commande (10), apres avoir delivre a la ligne de commande (3) une sortie de niveau BAS lorsque cette ligne de commande (3) est au niveau HAUT, et arrete de



delivrer a la ligne de commande (3) la sortie de niveau BAS  
apres  
avoir termine l'envoi de trame de donnees ou de trame vide,  
l'ordre d'envoi est predetermine pour les cibles (11, 12),  
les  
cibles (11, 12) avec une demande d'envoi de trame de donnees  
commencent a delivrer a la ligne de commande (3) une sortie de  
niveau BAS pendant que la ligne de commande (3) est maintenue  
au  
niveau BAS par le dispositif de commande (10), si la ligne  
de  
commande (3) reste au niveau BAS meme si le dispositif de  
commande  
(10) arrete de delivrer a la ligne de commande (3) la sortie de  
niveau BAS apres avoir termine l'envoi de trame de donnees ou  
de  
trame vide, le dispositif de commande (10) delivre en sortie  
l'impulsion d'horloge d'envoi jusqu'a ce que la ligne de  
commande  
(3) soit au niveau HAUT, et la cible (11, 12) delivrant a la  
ligne  
de commande (3) une sortie de niveau BAS commence la  
transmission  
d'une trame de donnees en synchronisation avec l'impulsion  
d'horloge d'envoi, apres que le dispositif de commande (10) et  
toutes les cibles (11, 12) anterieures dans l'ordre d'envoi  
auront  
acheve l'envoi de trame de donnees ou de trame vide, et arrete  
de  
delivrer une sortie de niveau BAS a la ligne de commande (3)  
apres  
qu'elle a termine la transmission de trame de donnees.

#### United States

Publication No. US 5600803 A (Update 199711 E)

Publication Date: 19970204

**\*\*Data transfer system and method using data bus having bi-directional serial data line, clock line and bi-directional control line.\*\***

Assignee: Matsushita Electric Industrial Co., Ltd., Osaka, JP (MATU)

Inventor: Iitsuka, Hiroyuki, Katano, JP

Nishimura, Takuya, Katano, JP

Agent: Wenderoth, Lind Ponack

Language: EN (32 pages, 13 drawings)

Application: US 1994242441 A 19940513 (Local application)

Priority: JP 1993112860 A 19930514

Original IPC: G06F-13/00(A)

Current IPC: G06F-13/00(A)

Original US Class (main): 395287

Original US Class (secondary): 395856 395855 395850 395200.05 364222.2  
364283.3 364238.3 364DIG.1

Original Abstract: A data transfer system including a plurality of microcomputers connected to a serial data bus in parallel to each other, one of which is designated as a controller and remainder as targets each having its own sending order of data allocated

beforehand,

in which the controller asserts a control line of the serial data bus

to start a data frame or a null frame sending to a data line of the serial data bus and each target outputs an output for asserting the control line while the control line is kept in an asserted state by the

controller if it has a data frame sending request, and outputs a data

frame or a null frame when its sending order has come.

Claim:

1. A data transfer method for serially sending data frames between one controller and one or more targets for which the order of sending

data is predetermined, comprising the steps of:

\* connecting said one controller and said one or more targets with a

bidirectional data line for serially sending data constituting at

least one data frame in a transfer unit, a clock line for intermittently supplying unit clocks required for sending one transfer unit of data at a predetermined output timing from

said

controller to said one or more targets, and a bidirectional control line;

\* said controller sending a data frame or null frame through said bidirectional data line in synchronization with said unit

clocks

after beginning a LOW level output to said control line when said

control line is HIGH;

\* each target having a transmission request of a data frame beginning a

LOW level output to said control line while said control line is

held LOW by said controller;

\* said controller outputting said unit clocks after resuming a LOW level output to said control line if said control line

remains

LOW even though said controller stops a LOW output to said control line at an output timing of said unit clock after finishing transmission of the data frame or null frame; and

\* said each target, outputting a LOW level to said control line, beginning transmission of a data frame in synchronization

with

said unit clock after completion of transmission of the data frame or null frame by said controller and all targets having

a

priority in the sending order which is higher than that of

the

pertinent target, and stopping a LOW output to said control

line

by the next output timing of a unit clock after completing transmission of the data frame.

?